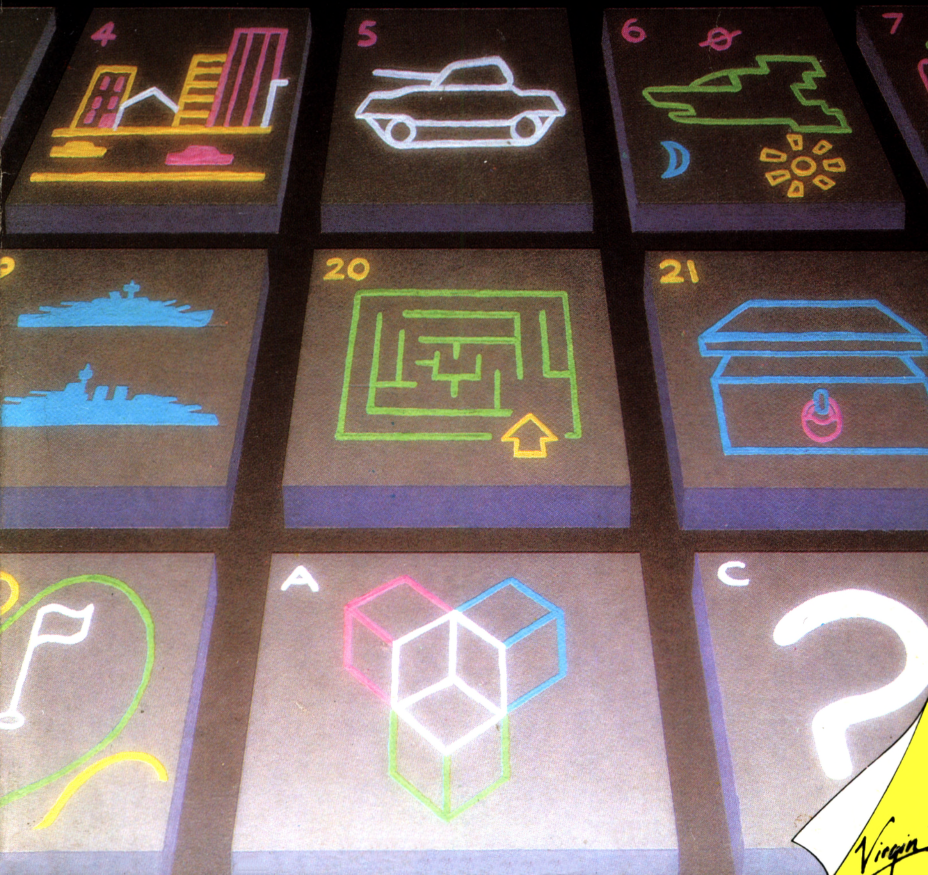
The Virgin Computer Games Series

GAMES FOR YOUR ZX81

£££££'s of entertaining games for only £2.95



GAMES FOR YOUR ZX81

Mark Charlton

Virgin Books

First published in Great Britain in 1983 by Virgin Books Ltd,
61-63 Portobello Road,London W11 3DD.

ISBN 0 907080 82 0

Printed and bound in Great Britain by Richard Clay (The Chaucer Press) Ltd, Suffolk.

Production services by Book Production Consultants, Cambridge.

Designed by Ray Hyden.

Illustrated by Sue Walliker.

Typeset by Portobello Typesetting.

Distributed by Hamlyn Paperbacks.

TIM HARTNELL — SERIES EDITOR

Tim Hartnell is a leading computer expert and journalist, who has contributed extensively to the Technical Consumer Press. He is also the author of several books including *Getting Acquainted With Your ZX81*, *Let Your BBC Micro Teach You to Program* and *Programming Your ZX Spectrum.*

MARK CHARLTON — THE AUTHOR

Mark Charlton has also written *The Gateway Guide to the ZX80 and ZX81* published by Interface (UK) and Creative Computing (US). He is a regular columnist for *Interface* magazine.

SUE WALLIKER — THE ILLUSTRATOR

Sue Walliker is a freelance illustrator.

ACKNOWLEDGEMENTS

The author would especially like to thank Paul Toland, Dilwyn Jones, Nick Wilson, Neil Streeter, Graham Charlton, Tim Rogers, Chris Callender, Clive Gifford, Alistair Miller and Jim Archer, who have all contributed material to this book.

CONTENTS

MOVING GRAPHICS GAMES

THE ZX81 FIGHTS BACK

MIND BENDERS

SELF DISCOVERY

THE ZX81 TAKES CONTROL

THE ZX81 LENDS A HAND

Introduction

Your computer is waiting to challenge you. Moving graphics games, brain stretchers, word games and puzzles are all here and ready to entertain you.

A wide variety of games are included in this book. The programs have been written by some of the most talented young programmers working in this country at the moment, and represent a variety of approaches to solving programming problems.

An examination of the listings should teach you many tricks and techniques to apply to your own programming. And once you have mastered the programs in their present form, you might want to try your hand at improving them. There is no such thing as a 'perfect program', so these games are sure to benefit from your programming skill.

All that now remains is for you to turn the page and enter the programs. I can only hope that you enjoy playing the games as much as we did when preparing this volume.

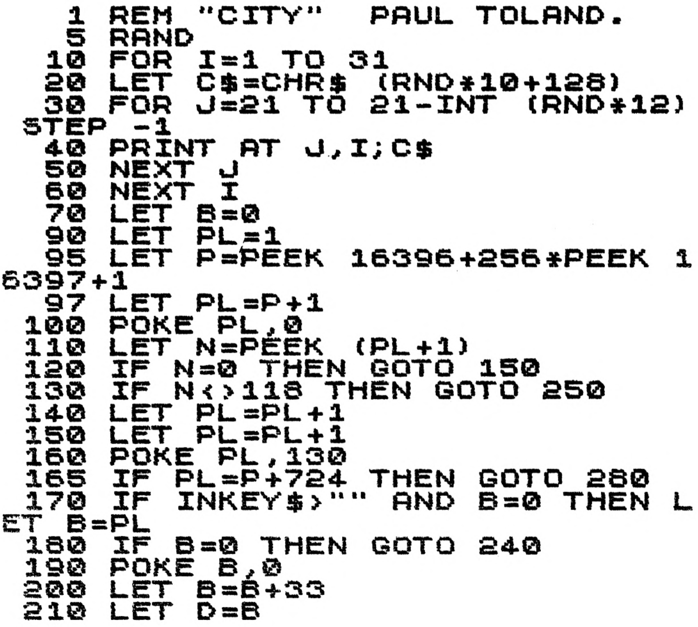
Tim Hartnell, series editor
London
March 1983

MOVING GRAPHICS GAMES

CITY

This challenging program is Paul Toland's version of the game in which a plane is descending on a large city, and must bomb away the skyscrapers in order to land. The program as listed is extremely hard (almost impossible!) to play, but it can be made slightly easier by changing the 12 in line 30 to a smaller number, such as 8. This changes the height of the skyscrapers. If you'd prefer to be flying over a village rather than a city, try 10 FOR 1 = 8 TO 25.

The city is different each time, and is generated at random. You fire your bomb by hitting any key except BREAK. When you have crashed (that is, finished) press any key except 'N' for a new game. To have a program which autostarts when loaded, save it by entering GOTO 340.

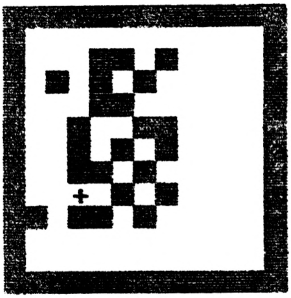


```
   1 REM "CITY"  PAUL TOLAND.
   5 RAND
  10 FOR I=1 TO 31
  20 LET C$=CHR$ (RND*10+128)
  30 FOR J=21 TO 21-INT (RND*12)
 STEP -1
  40 PRINT AT J,I;C$
  50 NEXT J
  60 NEXT I
  70 LET B=0
  90 LET PL=1
  95 LET P=PEEK 16396+256*PEEK 1
6397+1
  97 LET PL=P+1
 100 POKE PL,0
 110 LET N=PEEK (PL+1)
 120 IF N=0 THEN GOTO 150
 130 IF N<>118 THEN GOTO 250
 140 LET PL=PL+1
 150 LET PL=PL+1
 160 POKE PL,130
 165 IF PL=P+724 THEN GOTO 280
 170 IF INKEY$>"" AND B=0 THEN L
ET B=PL
 180 IF B=0 THEN GOTO 240
 190 POKE B,0
 200 LET B=B+33
 210 LET D=B
```

```
220 IF PEEK B>0 OR B>P+724 THEN
LET B=0
225 POKE D,28
227 LET I=1
230 POKE D,0
240 GOTO 100
250 POKE PL,131
251 POKE PL,132
252 LET I=1**1
253 POKE PL,133
254 LET I=1**1
255 POKE PL,134
256 LET I=1**1
257 POKE PL,135
258 LET I=1**1
259 POKE PL,23
260 PRINT AT 10,10;"YOU CRASHED
"
270 GOTO 290
280 PRINT AT 10,10;"YOU""R DOWN
IN ONE PIECE-"
290 PRINT ,"TRY AGAIN ?"
300 IF INKEY$="" THEN GOTO 300
310 IF INKEY$="N" THEN STOP
320 CLS
330 RUN
340 SAVE "CITY"
350 RUN
```

TRAP

This is a 1K moving graphics game written by Dilwyn Jones. You are the + sign wandering about the screen under the control of the '5' (left), '6' (down), '7' (up) and '8' (right) keys. You are awarded a point every time you move. The computer attempts to trap you by placing a blob next to you. The game is over when you are forced to move onto a blob. Your score is shown at the end.



```
  10 RAND
  20 LET A=0
  30 PRINT AT 14,0;"████████████
█";AT 2,0;"█████████████"
  40 FOR X=0 TO 10
  50 PRINT "█";TAB 12;"█"
  60 NEXT X
  70 LET X=INT (RND*11)+1
  80 LET Y=INT (RND*11)+3
  90 PRINT AT Y,X;
 100 IF PEEK (PEEK 16398+256*PEE
K 16399)=128 THEN GOTO 190
 110 PRINT "+";AT Y+(RND<.5)-(RN
D<.5),X+(RND<.5)-(RND<.5);"█" AN
D PEEK (PEEK 16398+256*PEEK 1639
9)<>21
 120 LET A=A+1
 130 LET A$=INKEY$
 140 IF A$<"5" OR A$>"8" THEN GO
TO 130
 150 PRINT AT Y,X;" "
 160 LET X=X+(A$="8")-(A$="5")
 170 LET Y=Y+(A$="6")-(A$="7")
 180 GOTO 90
 190 PRINT A
```

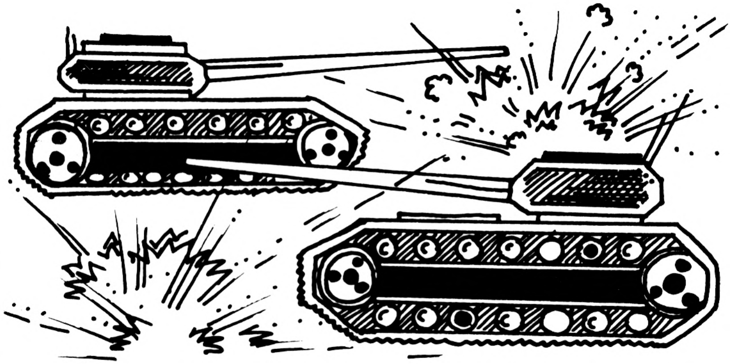
TANK ASSAULT

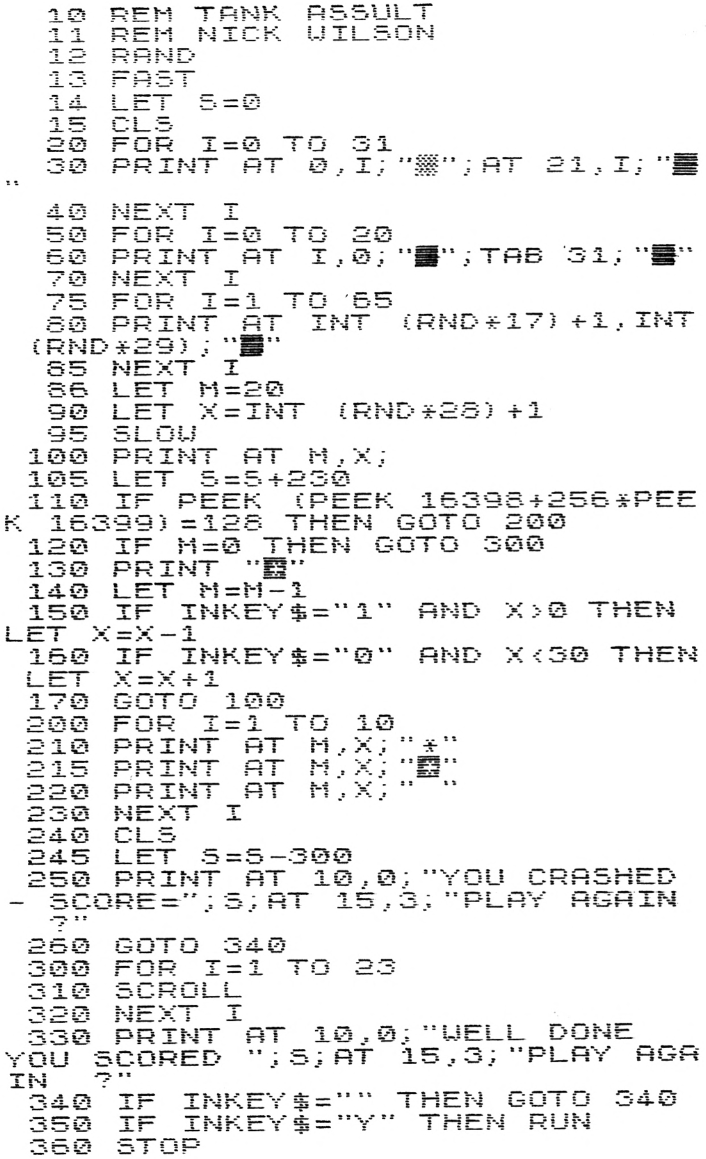
In this game, written for the 16K ZX81 by Nick Wilson, you have to drive a 'tank' from the bottom of the screen to the top, avoiding the black squares. These are bombs primed to blow you up. When you RUN the game, you will see the Assault Course printed on the screen, inside an explosive fence.

The tank (an inverse asterisk) will move up from the start of the game, and you have to steer it right (with the zero key) or left (with the one key). There are some 60 bombs in the game as listed, but you can easily change this by altering the 65 in line 75 to the number of your choice, with lower numbers making the game easier.

At the end, when you reach the top of the screen, you will be given a 'Tank Commando' rating. You'll find it a pretty fast-moving game and you may wish to slow it down the first few times you play it by adding 107 LET G = SIN PI. The graphics character in line 30 is from the H key.

Lines 12 to 85 print out the course and border; 100 to 170 move the tank up and to the left or right; 150 and 160 read the keyboard and move the tank; 200 to 230 create an explosion when the tank crashes, and lines 245 to 360 end the game and give you your rating.





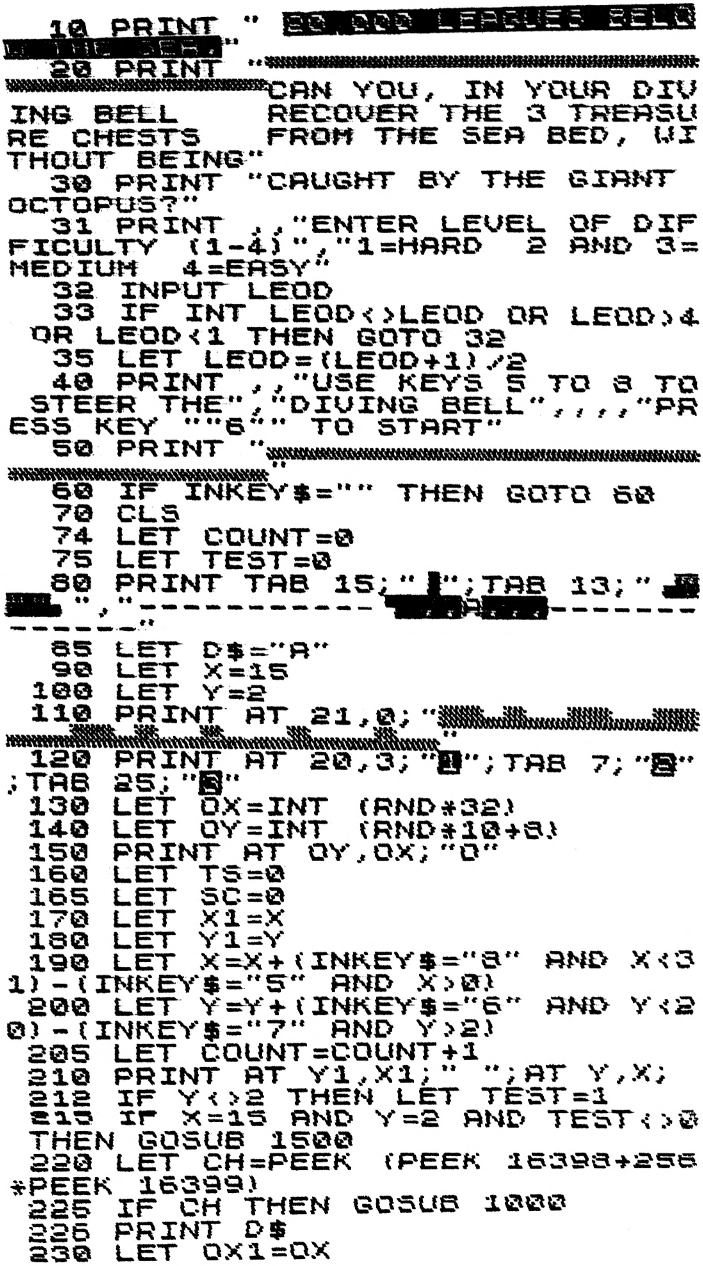
```
   10 REM TANK ASSULT
   11 REM NICK WILSON
   12 RAND
   13 FAST
   14 LET S=0
   15 CLS
   20 FOR I=0 TO 31
   30 PRINT AT 0,I;"▒";AT 21,I;"█
"
   40 NEXT I
   50 FOR I=0 TO 20
   60 PRINT AT I,0;"█";TAB 31;"█"
   70 NEXT I
   75 FOR I=1 TO 65
   80 PRINT AT INT (RND*17)+1,INT
 (RND*29);"█"
   85 NEXT I
   86 LET M=20
   90 LET X=INT (RND*28)+1
   95 SLOW
  100 PRINT AT M,X;
  105 LET S=S+230
  110 IF PEEK (PEEK 16398+256*PEE
K 16399)=128 THEN GOTO 200
  120 IF M=0 THEN GOTO 300
  130 PRINT "▣"
  140 LET M=M-1
  150 IF INKEY$="1" AND X>0 THEN
LET X=X-1
  160 IF INKEY$="0" AND X<30 THEN
 LET X=X+1
  170 GOTO 100
  200 FOR I=1 TO 10
  210 PRINT AT M,X;"*"
  215 PRINT AT M,X;"▣"
  220 PRINT AT M,X;" "
  230 NEXT I
  240 CLS
  245 LET S=S-300
  250 PRINT AT 10,0;"YOU CRASHED
- SCORE=";S;AT 15,3;"PLAY AGAIN
 ?"
  260 GOTO 340
  300 FOR I=1 TO 23
  310 SCROLL
  320 NEXT I
  330 PRINT AT 10,0;"WELL DONE
YOU SCORED ";S;AT 15,3;"PLAY AGA
IN ?"
  340 IF INKEY$="" THEN GOTO 340
  350 IF INKEY$="Y" THEN RUN
  360 STOP
```

20,000 LEAGUES BELOW THE SEA

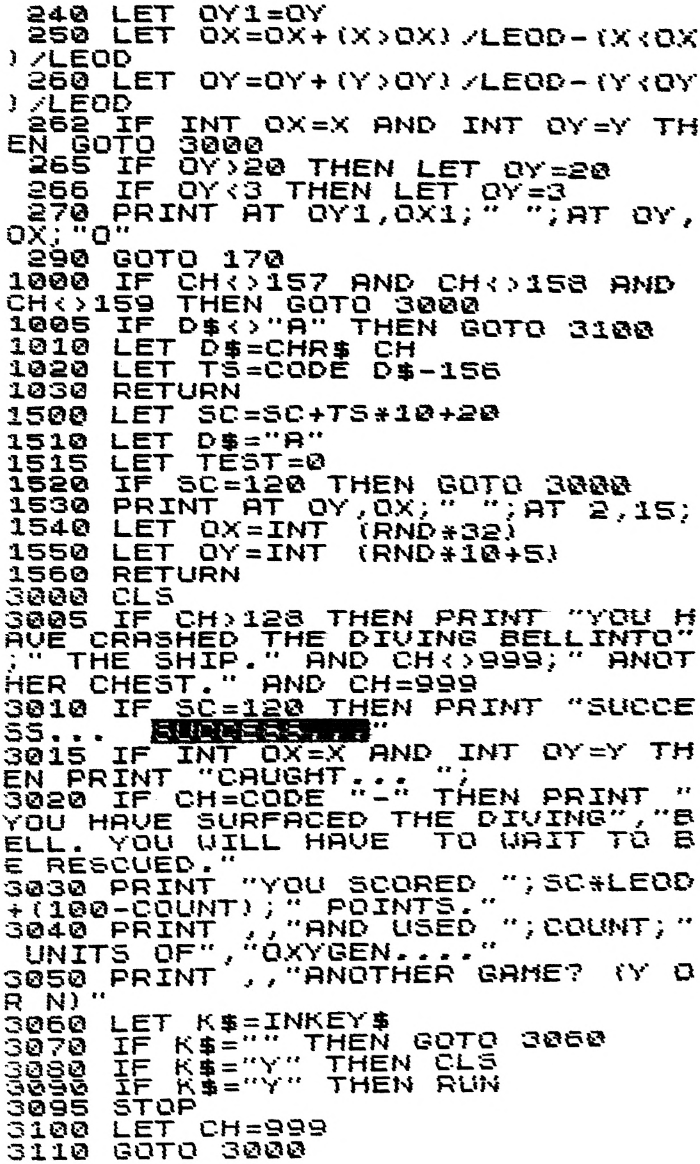
One problem with the ZX81 is that the graphics are rather slow when programs are written in BASIC, so that animated games become difficult to program. Space Invaders, for example, loses some of its interest if the invaders move only once every ten seconds or so.

If you are staying in BASIC, and not resorting to machine code, the only way to ensure graphics move at a realistic pace is to keep the program as short as possible and move as few things around the screen as possible. This is what Neil Streeter has done in this program. The only things that move are the diving bell and the octopus, each represented by a single character, which speeds up the printing.

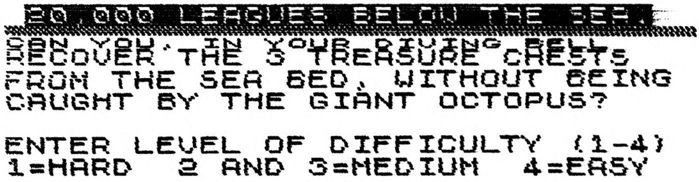
The main movement subroutine is between lines 170 and 290 and contains as little as possible (the tests for a key being pressed, or a treasure chest, or an octopus). Even the score is left until the end of the game, as are all messages. If you try and keep any movement subroutines as short as possible, and as close to the beginning of the program as you can, then you should find that a fairly acceptable speed of play can be achieved, as this program shows. The characters in line 80 are graphics 8, graphics 3, three inverse “”, graphics 4, eleven -, graphics 2, inverse space, two inverse “.”, A, three inverse “.”, and 13 - to finish.



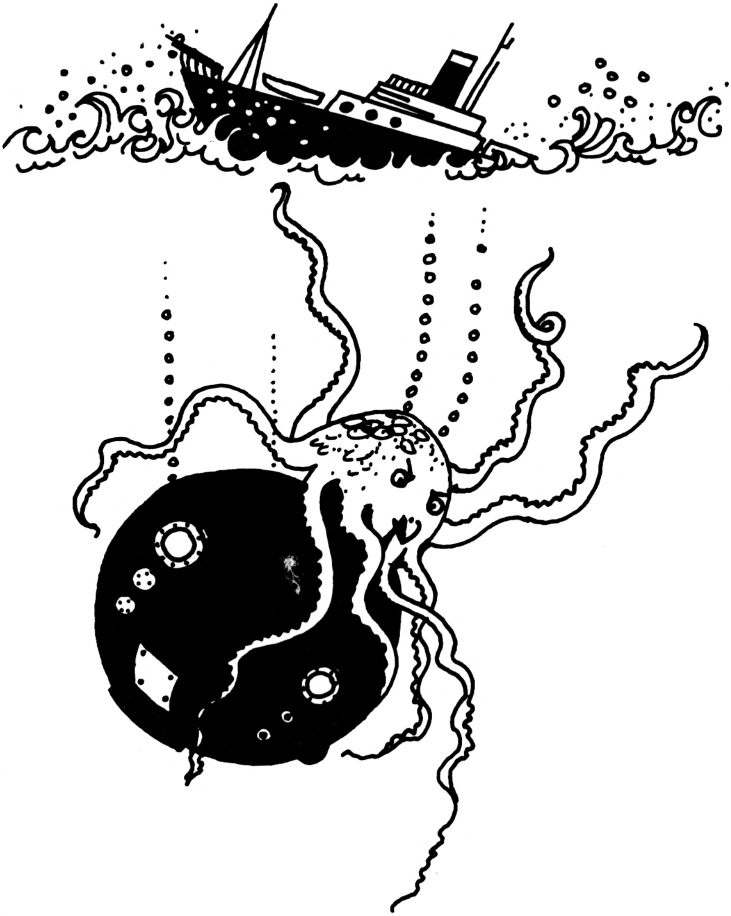
```
   10 PRINT " 20,000 LEAGUES BELO
W THE SEA."
   20 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒CAN YOU, IN YOUR DIV
ING BELL   RECOVER THE 3 TREASU
RE CHESTS  FROM THE SEA BED, WI
THOUT BEING"
   30 PRINT "CAUGHT BY THE GIANT
OCTOPUS?"
   31 PRINT ,,"ENTER LEVEL OF DIF
FICULTY (1-4)","1=HARD  2 AND 3=
MEDIUM  4=EASY"
   32 INPUT LEOD
   33 IF INT LEOD<>LEOD OR LEOD>4
 OR LEOD<1 THEN GOTO 32
   35 LET LEOD=(LEOD+1)/2
   40 PRINT ,,"USE KEYS 5 TO 8 TO
 STEER THE","DIVING BELL",,,,"PR
ESS KEY ""6"" TO START"
   50 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒"
   60 IF INKEY$="" THEN GOTO 60
   70 CLS
   74 LET COUNT=0
   75 LET TEST=0
   80 PRINT TAB 15;"▐";TAB 13;"▄█
█▙","-----------▄█▄▄▄▙▄▄▄-------
------"
   85 LET D$="A"
   90 LET X=15
  100 LET Y=2
  110 PRINT AT 21,0;"▒▒▒▒▄▄▒▒▄▄▄▄▒▒▄▄▒▒
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
  120 PRINT AT 20,3;"▣";TAB 7;"▣"
;TAB 25;"▣"
  130 LET OX=INT (RND*32)
  140 LET OY=INT (RND*10+8)
  150 PRINT AT OY,OX;"O"
  160 LET TS=0
  165 LET SC=0
  170 LET X1=X
  180 LET Y1=Y
  190 LET X=X+(INKEY$="8" AND X<3
1)-(INKEY$="5" AND X>0)
  200 LET Y=Y+(INKEY$="6" AND Y<2
0)-(INKEY$="7" AND Y>2)
  205 LET COUNT=COUNT+1
  210 PRINT AT Y1,X1;" ";AT Y,X;
  212 IF Y<>2 THEN LET TEST=1
  215 IF X=15 AND Y=2 AND TEST<>0
 THEN GOSUB 1500
  220 LET CH=PEEK (PEEK 16398+256
*PEEK 16399)
  225 IF CH THEN GOSUB 1000
  226 PRINT D$
  230 LET OX1=OX
```



```
 240 LET OY1=OY
 250 LET OX=OX+(X>OX)/LEOD-(X<OX
)/LEOD
 260 LET OY=OY+(Y>OY)/LEOD-(Y<OY
)/LEOD
 262 IF INT OX=X AND INT OY=Y TH
EN GOTO 3000
 265 IF OY>20 THEN LET OY=20
 266 IF OY<3 THEN LET OY=3
 270 PRINT AT OY1,OX1;" ";AT OY,
OX;"O"
 290 GOTO 170
1000 IF CH<>157 AND CH<>158 AND
CH<>159 THEN GOTO 3000
1005 IF D$<>"A" THEN GOTO 3100
1010 LET D$=CHR$ CH
1020 LET TS=CODE D$-156
1030 RETURN
1500 LET SC=SC+TS*10+20
1510 LET D$="A"
1515 LET TEST=0
1520 IF SC=120 THEN GOTO 3000
1530 PRINT AT OY,OX;" ";AT 2,15;
1540 LET OX=INT (RND*32)
1550 LET OY=INT (RND*10+5)
1560 RETURN
3000 CLS
3005 IF CH>128 THEN PRINT "YOU H
AVE CRASHED THE DIVING BELLINTO"
;" THE SHIP." AND CH<>999;" ANOT
HER CHEST." AND CH=999
3010 IF SC=120 THEN PRINT "SUCCE
SS...  SUCCESS..."
3015 IF INT OX=X AND INT OY=Y TH
EN PRINT "CAUGHT... ";
3020 IF CH=CODE "-" THEN PRINT "
YOU HAVE SURFACED THE DIVING","B
ELL. YOU WILL HAVE  TO WAIT TO B
E RESCUED."
3030 PRINT "YOU SCORED ";SC*LEOD
+(100-COUNT);" POINTS."
3040 PRINT ,,"AND USED ";COUNT;"
 UNITS OF","OXYGEN...."
3050 PRINT ,,"ANOTHER GAME? (Y O
R N)"
3060 LET K$=INKEY$
3070 IF K$="" THEN GOTO 3060
3080 IF K$="Y" THEN CLS
3090 IF K$="Y" THEN RUN
3095 STOP
3100 LET CH=999
3110 GOTO 3000
```



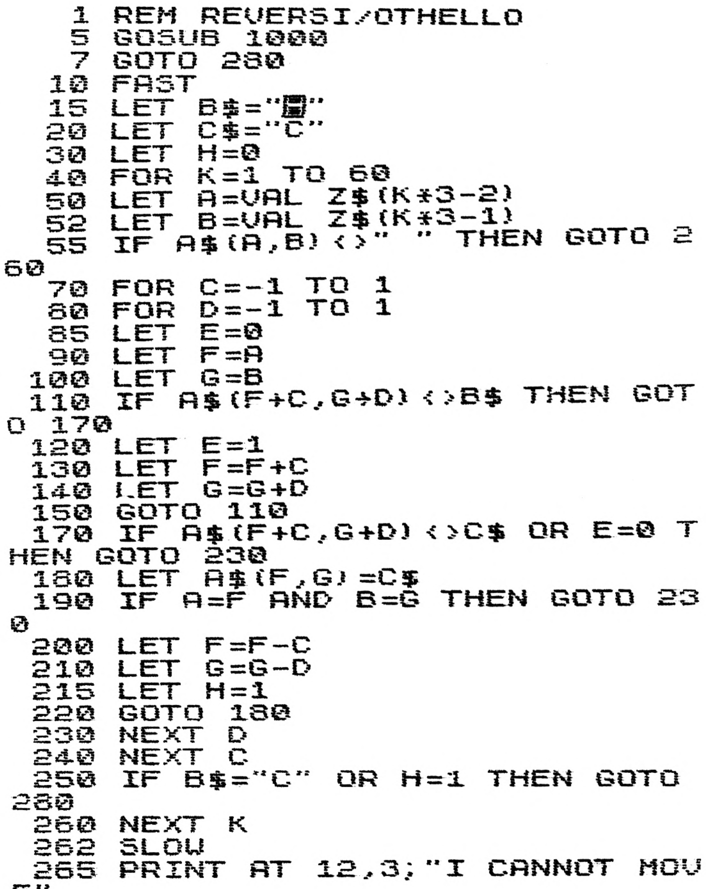
```
20,000 LEAGUES BELOW THE SEA
CAN YOU, IN YOUR DIVING BELL
RECOVER THE 3 TREASURE CHESTS
FROM THE SEA BED, WITHOUT BEING
CAUGHT BY THE GIANT OCTOPUS?

ENTER LEVEL OF DIFFICULTY (1-4)
1=HARD  2 AND 3=MEDIUM  4=EASY
```



THE ZX81 FIGHTS BACK

REVERSI

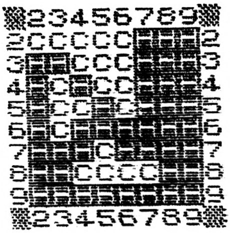
This popular board game — often called Othello — has attracted many computer programmers. Graham Charlton is one such programmer. He has written a number of Reversi programs, including the core of the one given away on a flexidisc with an issue of *Your Computer* magazine.



```
   1 REM REVERSI/OTHELLO
   5 GOSUB 1000
   7 GOTO 280
  10 FAST
  15 LET B$="▦"
  20 LET C$="C"
  30 LET H=0
  40 FOR K=1 TO 60
  50 LET A=VAL Z$(K*3-2)
  52 LET B=VAL Z$(K*3-1)
  55 IF A$(A,B)<>" " THEN GOTO 2
60
  70 FOR C=-1 TO 1
  80 FOR D=-1 TO 1
  85 LET E=0
  90 LET F=A
 100 LET G=B
 110 IF A$(F+C,G+D)<>B$ THEN GOT
O 170
 120 LET E=1
 130 LET F=F+C
 140 LET G=G+D
 150 GOTO 110
 170 IF A$(F+C,G+D)<>C$ OR E=0 T
HEN GOTO 230
 180 LET A$(F,G)=C$
 190 IF A=F AND B=G THEN GOTO 23
0
 200 LET F=F-C
 210 LET G=G-D
 215 LET H=1
 220 GOTO 180
 230 NEXT D
 240 NEXT C
 250 IF B$="C" OR H=1 THEN GOTO
280
 260 NEXT K
 262 SLOW
 265 PRINT AT 12,3;"I CANNOT MOV
E"
```

```
  270 FOR Z=1 TO 100
  272 NEXT Z
  275 PRINT AT 12,3;"
  "

  280 SLOW
  285 PRINT AT 0,0;
  290 FOR A=1 TO 10
  310 PRINT A$(A)
  340 NEXT A
  350 IF B$="C" THEN GOTO 10
  360 PRINT AT 12,3;"ENTER NUMBER
  DOWN SIDE"
  362 INPUT A
  365 IF A<0 OR A>9 THEN GOTO 362
  366 PRINT AT 12,3;"
                    "

  367 IF A=0 AND K=61 THEN GOTO 1
 120
  368 IF A=0 THEN GOTO 10
  370 PRINT AT 12,3;"ENTER NUMBER
  ACROSS TOP"
  372 INPUT B
  375 IF B<0 OR B>9 THEN GOTO 372
  376 PRINT AT 12,3;"
                    "

  380 LET B$="C"
  390 LET C$="▣"
  400 GOTO 70
 1000 DIM A$(10,10)
 1005 LET K=0
 1006 LET B$="▣"
 1010 LET A$(1)="▒23456789▒"
 1020 LET A$(2)="2        2"
 1030 LET A$(3)="3        3"
 1040 LET A$(4)="4        4"
 1050 LET A$(5)="5   C▣   5"
 1060 LET A$(6)="6   ▣C   6"
 1070 LET A$(7)="7        7"
 1080 LET A$(8)="8        8"
 1090 LET A$(9)="9        9"
 1100 LET A$(10)="▒23456789▒"
 1102 LET Z$="22,29,92,99,94,97,7
 2,79,42,49,24,27,74,77,44,47,75,
 76,46,45,54,67,57,64,25,26,95,96
 ,52,59,62,69,34,37,84,87,43,48,7
 3,78,35,36,85,86,53,58,63,68,23,
 28,93,98,32,39,82,89,33,88,83,38
  "
 ,
 1119 RETURN
 1120 PRINT AT 12,3;"END OF THE G
 AME"
 1125 LET C=0
 1126 LET H=0
 1130 FOR A=2 TO 9
 1140 FOR B=2 TO 9
```

```
1150 IF A$(A,B)="C" THEN LET C=C
+1
1160 IF A$(A,B)="▦" THEN LET H=H
+1
1170 NEXT B
1175 NEXT A
1180 IF H>C THEN PRINT ,,"YOU WI
N"
1190 IF H<C THEN PRINT ,,"I WIN"
1200 IF H=C THEN PRINT ,,"ITS A
DRAW"
1210 PRINT ,,"I SCORED ";C
1220 PRINT ,,"YOU SCORED ";H
1230 PRINT ,,"DO YOU WANT ANOTHE
R GAME (Y/N)?"
1235 IF INKEY$="" THEN GOTO 1235
1270 IF INKEY$="N" THEN STOP
1280 CLS
1290 RUN
```



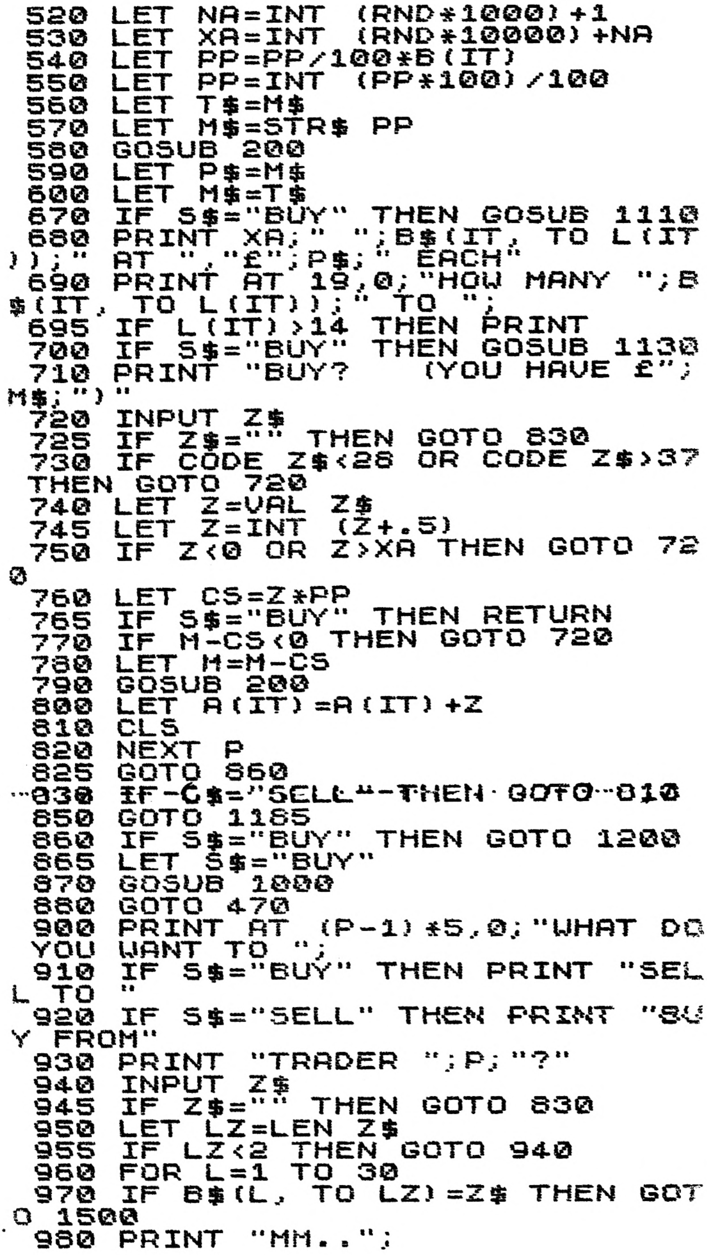
```
    END OF THE GAME

 YOU WIN

 I SCORED 20

 YOU SCORED 44

 DO YOU WANT ANOTHER GAME (Y/N)?
```

STAR-TRADER

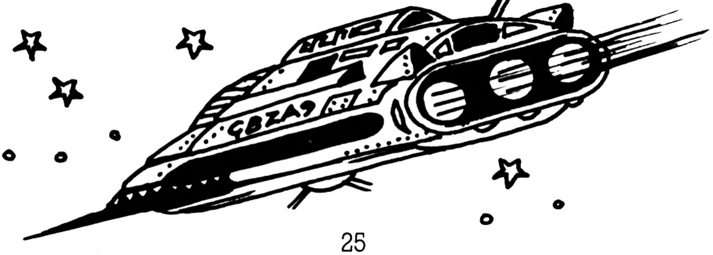
You are speeding through the space lanes, a merchant trader plying the galactic markets. You begin your journey with a number of goods and some intergalactic money, and attempt to return to port with more goods. Your cargo will be sold for twice the standard rate when you arrive, and you will receive ten per cent of the profits. If you don't want to buy or sell from any particular trader, just press ENTER.

Out there in space you will meet other star-traders on their way to different planetary systems and they will offer to buy and sell goods at cut prices. The aim of the game is to stockpile expensive goods, with money just the means to that end. Star-Trader was written by Tim Rogers.

```
  10 LET A$="EX-DICTATORS,WELLIN
GTONS,PAPER BAGS,PLASTIC BRICKS,
TONNES GRAIN,MISSILES,RUGBY BOOT
S,SMARTIES,BARS REJECT SOAP,TONS
 ORANGE PEEL,KG ICING,CLOTHES PE
GS,TYRES,ZINC HUB CAPS,GAL. SILI
CON INK,SACKS COAL DUST,METAL IN
GOTS,TEST TUBES,DOOR KNOBS,FISH,
TV SETS,BADGES,C-60 CASSETTES,SL
IPPERS,TIN NAILS,BRICKS,ASPIDIST
RAS,LIGHT BULBS,GLOVES,PEN TOPS"
  15 FAST
  20 DIM B$(30,16)
  25 LET F=100
  30 DIM L(30)
  40 LET A=1
  45 GOSUB 125
  50 FOR B=1 TO LEN A$
  70 IF A$(B)="," THEN GOSUB F
  80 LET C$=C$+A$(B)
  90 NEXT B
  95 GOTO F
 100 LET B$(A)=C$
 110 LET L(A)=LEN C$
 115 LET A=A+1
 120 LET B=B+1
 125 LET C$=""
 130 IF A<=30 THEN RETURN
 135 IF F=180 THEN GOTO 230
 140 LET A$="20,20,.1,.5,50,1000
,20,.2,.5,1,1,5,10,45,4,5,600,2.
5,5,6,100,1,4,9,3,70,5,6,15,.4"
 150 DIM B(30)
 160 LET F=180
 170 GOTO 40
 180 LET B(A)=VAL C$
 190 GOTO 115
 200 LET M=INT (M*100)/100
 202 IF M$<>STR$ PP THEN LET M$=
STR$ M
 205 IF LEN M$<2 THEN GOTO 220
 210 IF M$(LEN M$-1)="." THEN LE
T M$=M$+"0"
 220 RETURN
 230 DIM A(30)
 240 LET IA=INT (RND*10000)+1
 250 LET M=IA
 260 FOR B=1 TO 30
 270 IF RND>.85 THEN LET A(B)=IN
T (RND*10000)+1
 280 IF A(B)>0 THEN LET IA=IA+B(
B)*2
 290 NEXT B
 300 LET Y=2100+INT (RND*3000)
 305 LET PP=0
 310 SLOW
 315 LET M$=STR$ M
 320 GOSUB 200
 330 FOR Y=Y TO Y+20
 340 CLS
 350 PRINT TAB 12;"YEAR ";Y
 355 LET P=3
 360 PRINT "********************
************"
 370 PRINT "MONEY £";
 380 FOR B=1 TO LEN M$
 390 PRINT CHR$ (CODE M$(B)+128)
;
 400 NEXT B
 405 PRINT
 410 FOR B=1 TO 30
 420 IF A(B)=0 THEN GOTO 440
 425 PRINT A(B);TAB 6;B$(B)
 430 LET P=P+1
 435 IF P>=18 THEN SCROLL
 440 NEXT B
 450 LET S$="SELL"
 460 GOSUB 1000
 470 FOR P=1 TO NT
 475 IF RND<.47 THEN GOTO 900
 480 PRINT AT (P-1)*5,0;"TRADER
";P;
 490 PRINT " WISHES TO ";S$
 500 LET IT=INT (RND*30)+1
 510 LET PP=INT (RND*100)+1
```



```
 520 LET NA=INT (RND*1000)+1
 530 LET XA=INT (RND*10000)+NA
 540 LET PP=PP/100*B(IT)
 550 LET PP=INT (PP*100)/100
 560 LET T$=M$
 570 LET M$=STR$ PP
 580 GOSUB 200
 590 LET P$=M$
 600 LET M$=T$
 670 IF S$="BUY" THEN GOSUB 1110
 680 PRINT XA;" ";B$(IT, TO L(IT
));" AT ","£";P$;" EACH"
 690 PRINT AT 19,0;"HOW MANY ";B
$(IT, TO L(IT));" TO ";
 695 IF L(IT)>14 THEN PRINT
 700 IF S$="BUY" THEN GOSUB 1130
 710 PRINT "BUY?    (YOU HAVE £";
M$;")"
 720 INPUT Z$
 725 IF Z$="" THEN GOTO 830
 730 IF CODE Z$<28 OR CODE Z$>37
 THEN GOTO 720
 740 LET Z=VAL Z$
 745 LET Z=INT (Z+.5)
 750 IF Z<0 OR Z>XA THEN GOTO 72
0
 760 LET CS=Z*PP
 765 IF S$="BUY" THEN RETURN
 770 IF M-CS<0 THEN GOTO 720
 780 LET M=M-CS
 790 GOSUB 200
 800 LET A(IT)=A(IT)+Z
 810 CLS
 820 NEXT P
 825 GOTO 860
 830 IF C$="SELL" THEN GOTO 810
 850 GOTO 1165
 860 IF S$="BUY" THEN GOTO 1200
 865 LET S$="BUY"
 870 GOSUB 1000
 880 GOTO 470
 900 PRINT AT (P-1)*5,0;"WHAT DO
 YOU WANT TO ";
 910 IF S$="BUY" THEN PRINT "SEL
L TO "
 920 IF S$="SELL" THEN PRINT "BU
Y FROM"
 930 PRINT "TRADER ";P;"?"
 940 INPUT Z$
 945 IF Z$="" THEN GOTO 830
 950 LET LZ=LEN Z$
 955 IF LZ<2 THEN GOTO 940
 960 FOR L=1 TO 30
 970 IF B$(L, TO LZ)=Z$ THEN GOT
O 1500
 980 PRINT "MM..";
 990 NEXT L
 995 GOTO 1510
1000 LET NT=INT (RND*5)
1010 PRINT AT 19,0;"THERE ARE ";
NT;" TRADERS WISHING TO"
1020 PRINT S$;" SOME GOODS ";
1030 IF S$="SELL" THEN PRINT "TO
";
1040 IF S$="BUY" THEN PRINT "FRO
M";
1050 PRINT " YOU"
1060 PRINT AT 21,18;"PRESS ANY K
EY"
1070 PAUSE 4E4
1080 POKE 16437,255
1090 CLS
1100 RETURN
1110 PRINT NA;" TO ";
1120 RETURN
1130 PRINT "SELL?   (YOU HAVE ";
A(IT);" ";B$(IT, TO L(IT));")"
1140 GOSUB 720
1150 IF Z<NA OR A(IT)<Z THEN GOS
UB 720
1160 LET A(IT)=A(IT)-Z
1170 LET M=M+CS
1180 GOSUB 200
1185 CLS
1190 NEXT P
1200 NEXT Y
1210 LET FA=M
1220 FOR B=1 TO 30
1230 PRINT AT 11,10;"YOU HAVE AR
RIVED"
1240 IF A(B)>0 THEN LET FA=FA+B(
B)*2
1250 PRINT AT 11,10;"YOU HAVE AR
RIVED"
1260 NEXT B
1270 CLS
1280 PRINT "YOU LEFT WITH £";IA;
" WORTH OF"
1290 PRINT "CARGO AND MONEY ABOA
RD"
1300 PRINT ,,"YOU HAVE NOW GOT £
";FA;" WORTH"
1310 PRINT "OF GOODS AND MONEY A
BOARD"
1320 LET PR=FA-IA
1330 IF SGN PR=-1 THEN LET P$="L
OSS"
1340 IF SGN PR>-1 THEN LET P$="P
ROFIT"
1350 PRINT ,,"YOU HAVE MADE A ";
P$;" OF £";ABS PR
1360 IF P$="PROFIT" THEN PRINT ,
,,,"YOU HAVE BEEN PAID £";INT (P
R*10)/100
1370 IF P$="LOSS" THEN PRINT ,,,
,"YOU HAVE BEEN","LOCKED UP AND
WILL NEVER SEE","THE LIGHT OF DA
Y AGAIN - LET","ALONE THE INSIDE
 OF ANOTHER","SPACE-SHIP"
1372 PAUSE 200
1375 POKE 16437,255
1380 IF P$="LOSS" THEN NEW
1390 PRINT ,,,,"YOUR EMPLOYERS H
AVE OFFERED YOU","ANOTHER CONTRA
CT - ACCEPT?"
1400 IF INKEY$<>"Y" AND INKEY$<>
"N" THEN GOTO 1400
1410 IF INKEY$="N" THEN STOP
1420 FAST
1430 GOTO 230
1500 CLS
1505 IF RND>.4 THEN GOTO 1600
1510 CLS
1520 IF L<31 THEN LET Z$=B$(L, T
O L(L))
1530 PRINT AT (P-1)*5,0;"TRADER
";P;" DOES NOT ";
1535 LET R=RND
1540 IF S$="SELL" AND R<.5 THEN
PRINT "HAVE ANY"
1550 IF S$="SELL" AND R>=.5 THEN
 PRINT "WANT TO","SELL ANY ";
1560 IF S$="BUY" THEN PRINT "WAN
T ANY"
1570 PRINT Z$
1580 PRINT AT 21,18;"PRESS ANY K
EY"
1585 PAUSE 4E4
1590 POKE 16437,255
1595 GOTO 830
1600 PRINT AT (P-1)*5,0;"TRADER
";P;" WISHES TO ";S$
1610 LET IT=L
1620 GOTO 510
2000 POKE 16389,76
2010 SAVE "SLAVES II"
2020 POKE 16389,128
2030 FAST
2040 GOTO 230
```



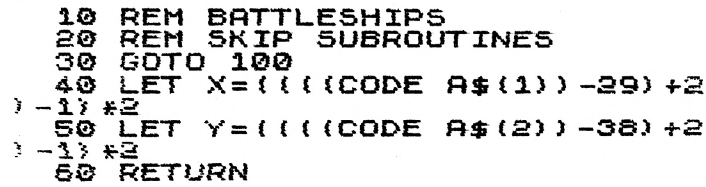
BATTLESHIPS

This game, by Nick Wilson, allows you to challenge your ZX81 to the game of Battleships, usually played with pen and paper. All the normal rules are followed.

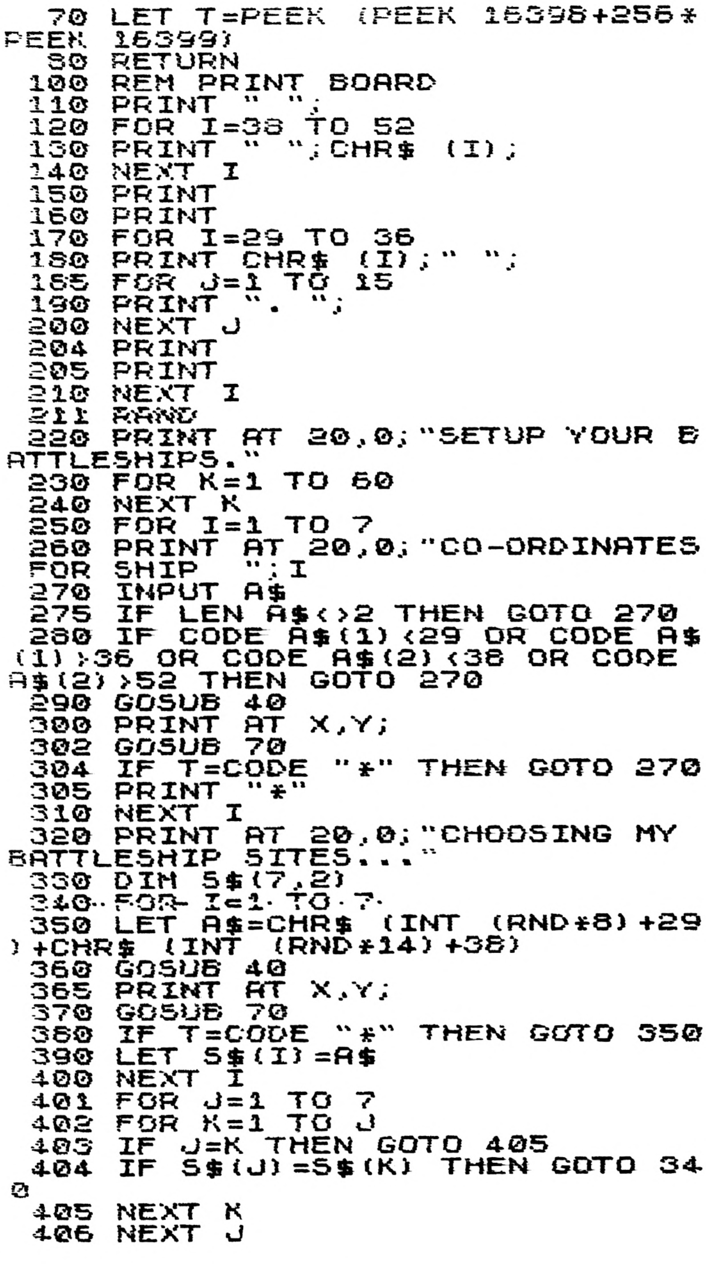
When you run the program the board will be printed, and you will be asked to enter the co-ordinates of your seven battleships, in row—column, number—letter form. The square designated is then checked to see if it is empty and, if so, the program prints your battleship there. The computer then sets up its own battleships, storing their locations in a string array so that it knows where they are ... but you do not.

After this initial procedure the program goes into a cycle, asking first for moves from you, then the computer, then back to you and so on, until all the ships on one side are destroyed. Your battleships are shown as asterisks, a destroyed ship as an inverse asterisk, an empty square that has been fired on as a colon, and an untouched square as a full stop.

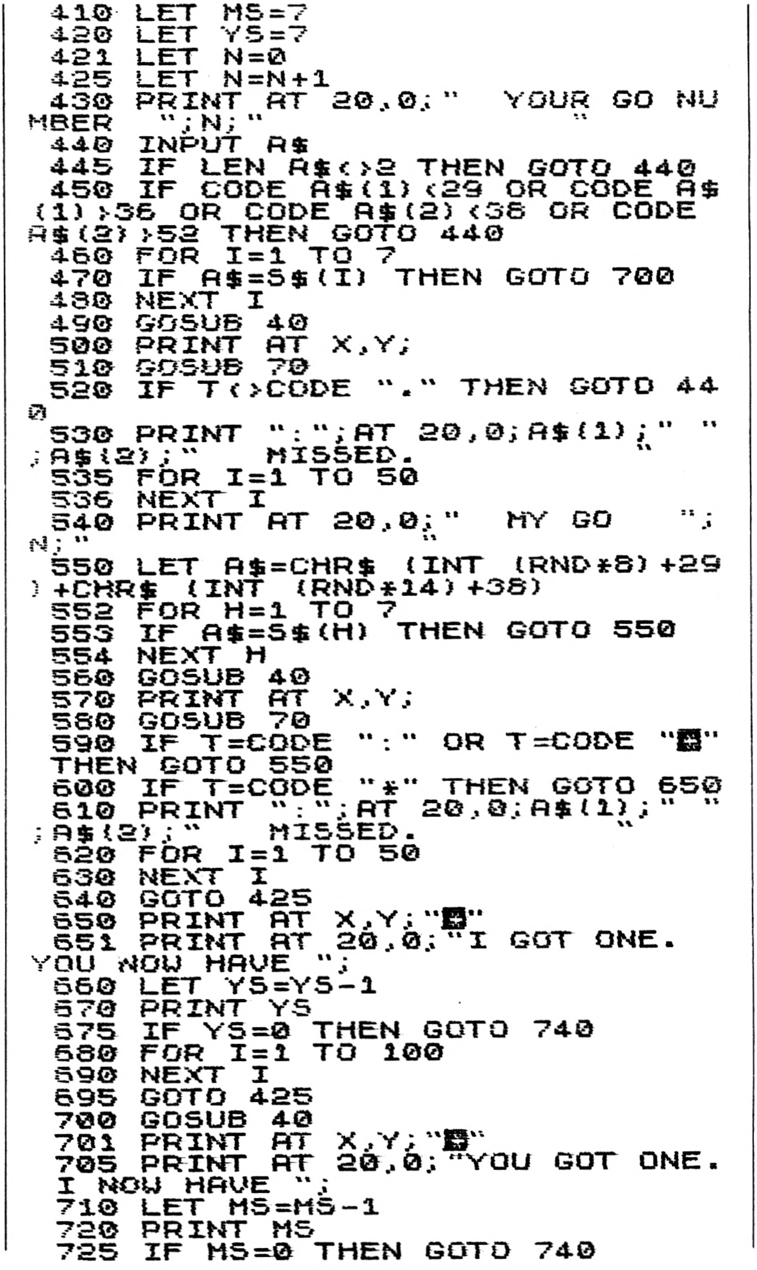
The game ends when the ships belonging either to you or to the computer are destroyed. The computer usually makes its moves very quickly at the start of the game, but it takes gradually longer as the game progresses and fewer vacant squares are left on the board. It is important to adhere closely to the program listing, as the screen is used as the sole memory for everything except the computer's ships. The number 7 in lines 250, 330, 401, 460 and 552 can be made lower, giving less ships, or higher, giving more. As listed, an average game takes from 25 to 40 minutes.



```
  10 REM BATTLESHIPS
  20 REM SKIP SUBROUTINES
  30 GOTO 100
  40 LET X=((((CODE A$(1))-29)+2
)-1)*2
  50 LET Y=((((CODE A$(2))-38)+2
)-1)*2
  60 RETURN
```

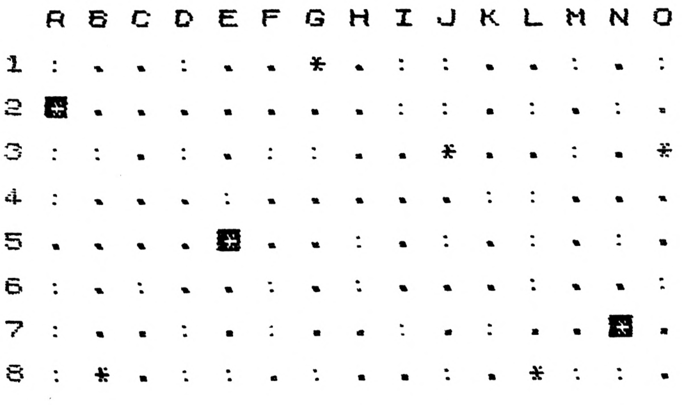


```
  70 LET T=PEEK (PEEK 16398+256*
PEEK 16399)
  80 RETURN
 100 REM PRINT BOARD
 110 PRINT " ";
 120 FOR I=38 TO 52
 130 PRINT " ";CHR$ (I);
 140 NEXT I
 150 PRINT
 160 PRINT
 170 FOR I=29 TO 36
 180 PRINT CHR$ (I);" ";
 185 FOR J=1 TO 15
 190 PRINT ". ";
 200 NEXT J
 204 PRINT
 205 PRINT
 210 NEXT I
 211 RAND
 220 PRINT AT 20,0;"SETUP YOUR B
ATTLESHIPS."
 230 FOR K=1 TO 60
 240 NEXT K
 250 FOR I=1 TO 7
 260 PRINT AT 20,0;"CO-ORDINATES
FOR SHIP  ";I
 270 INPUT A$
 275 IF LEN A$<>2 THEN GOTO 270
 280 IF CODE A$(1)<29 OR CODE A$
(1)>36 OR CODE A$(2)<38 OR CODE
A$(2)>52 THEN GOTO 270
 290 GOSUB 40
 300 PRINT AT X,Y;
 302 GOSUB 70
 304 IF T=CODE "*" THEN GOTO 270
 305 PRINT "*"
 310 NEXT I
 320 PRINT AT 20,0;"CHOOSING MY
BATTLESHIP SITES..."
 330 DIM S$(7,2)
 340 FOR I=1 TO 7
 350 LET A$=CHR$ (INT (RND*8)+29
)+CHR$ (INT (RND*14)+38)
 360 GOSUB 40
 365 PRINT AT X,Y;
 370 GOSUB 70
 380 IF T=CODE "*" THEN GOTO 350
 390 LET S$(I)=A$
 400 NEXT I
 401 FOR J=1 TO 7
 402 FOR K=1 TO J
 403 IF J=K THEN GOTO 405
 404 IF S$(J)=S$(K) THEN GOTO 34
0
 405 NEXT K
 406 NEXT J
```



```
 410 LET MS=7
 420 LET YS=7
 421 LET N=0
 425 LET N=N+1
 430 PRINT AT 20,0;"  YOUR GO NU
MBER  ";N;"                 "
 440 INPUT A$
 445 IF LEN A$<>2 THEN GOTO 440
 450 IF CODE A$(1)<29 OR CODE A$
(1)>36 OR CODE A$(2)<38 OR CODE
A$(2)>52 THEN GOTO 440
 460 FOR I=1 TO 7
 470 IF A$=S$(I) THEN GOTO 700
 480 NEXT I
 490 GOSUB 40
 500 PRINT AT X,Y;
 510 GOSUB 70
 520 IF T<>CODE "." THEN GOTO 44
0
 530 PRINT ":";AT 20,0;A$(1);"  "
;A$(2);"   MISSED.           "
 535 FOR I=1 TO 50
 536 NEXT I
 540 PRINT AT 20,0;"  MY GO    ";
N;"  "
 550 LET A$=CHR$ (INT (RND*8)+29
)+CHR$ (INT (RND*14)+38)
 552 FOR H=1 TO 7
 553 IF A$=S$(H) THEN GOTO 550
 554 NEXT H
 560 GOSUB 40
 570 PRINT AT X,Y;
 580 GOSUB 70
 590 IF T=CODE ":" OR T=CODE "▒"
 THEN GOTO 550
 600 IF T=CODE "*" THEN GOTO 650
 610 PRINT ":";AT 20,0;A$(1);"  "
;A$(2);"   MISSED.          "
 620 FOR I=1 TO 50
 630 NEXT I
 640 GOTO 425
 650 PRINT AT X,Y;"▒"
 651 PRINT AT 20,0;"I GOT ONE.
YOU NOW HAVE ";
 660 LET YS=YS-1
 670 PRINT YS
 675 IF YS=0 THEN GOTO 740
 680 FOR I=1 TO 100
 690 NEXT I
 695 GOTO 425
 700 GOSUB 40
 701 PRINT AT X,Y;"▒"
 705 PRINT AT 20,0;"YOU GOT ONE.
 I NOW HAVE ";
 710 LET MS=MS-1
 720 PRINT MS
 725 IF MS=0 THEN GOTO 740
```

```
730 GOTO 680
740 IF YS=0 THEN LET W$="I"
750 IF YS=0 THEN LET L$="YOU"
760 IF MS=0 THEN LET W$="YOU"
770 IF MS=0 THEN LET L$="I"
780 FOR I=1 TO 20
790 SCROLL
800 NEXT I
810 CLS
820 PRINT W$;" WIN."
830 PRINT
840 PRINT L$;" LOSE."
850 STOP
```



```
   A B C D E F G H I J K L M N O

1  : . . : . . * . : : . . : . :

2  ■ . . . . . . . : : . : . : .

3  : : . : . : : . . * . . : . *

4  : . . . : . . . . . : : . . .

5  . . . . ■ . . : . : . : . : .

6  : . : . . : . : . . . : . . :

7  : . . : . : . . : . : . . ■ .

8  : * . : : . : . . : . * : : .


YOU GOT ONE. I NOW HAVE 6
```

■....DESTROYED SHIP.

*...HUMAN SHIP

:...SQUARE THAT HAS BEEN FIRED

. ..UNTOUCHED SQUARE

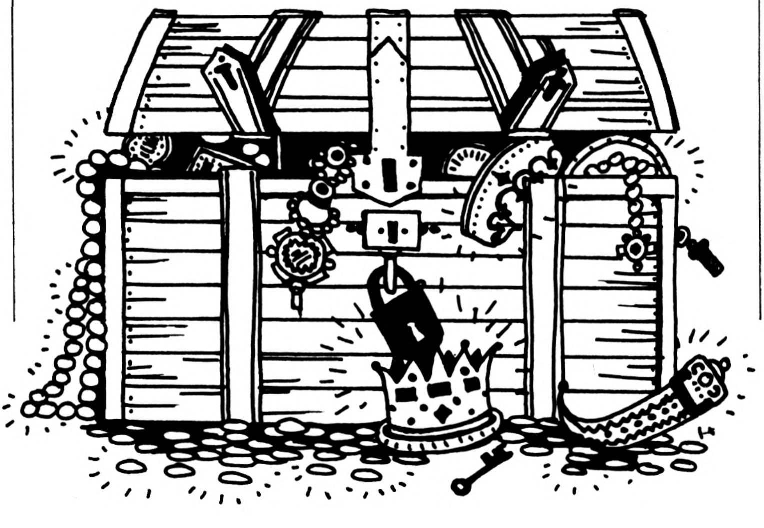
(THE COMPUTER/S SHIPS ARE INVISIBLE ON THE SCREEN)

TREASURE CHEST NIM

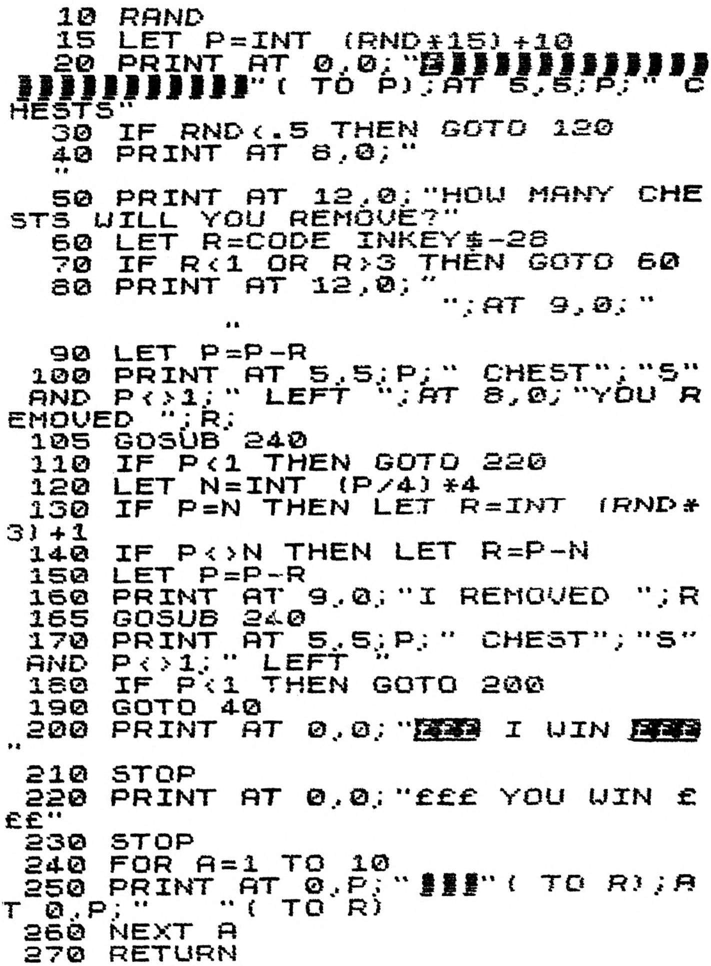
You and the ZX81 take it in turns to remove treasure chests from a long line of such chests. Every chest is empty except for the last, which contains jewels and gems to make you wealthy. The contents of the chest belong to the player who manages to take the last chest.

You can take one, two or three treasure chests away from the line, in an attempt to force the clever computer into allowing you to take the booty. The computer will usually win, though if you concentrate you may win a few games. You take away chests by touching the 1, 2 or 3 keys.

In this program, written by Dilwyn Jones, the computer always keeps you informed of how many it has removed, and there is both a numeric and a graphical display of the state of play.



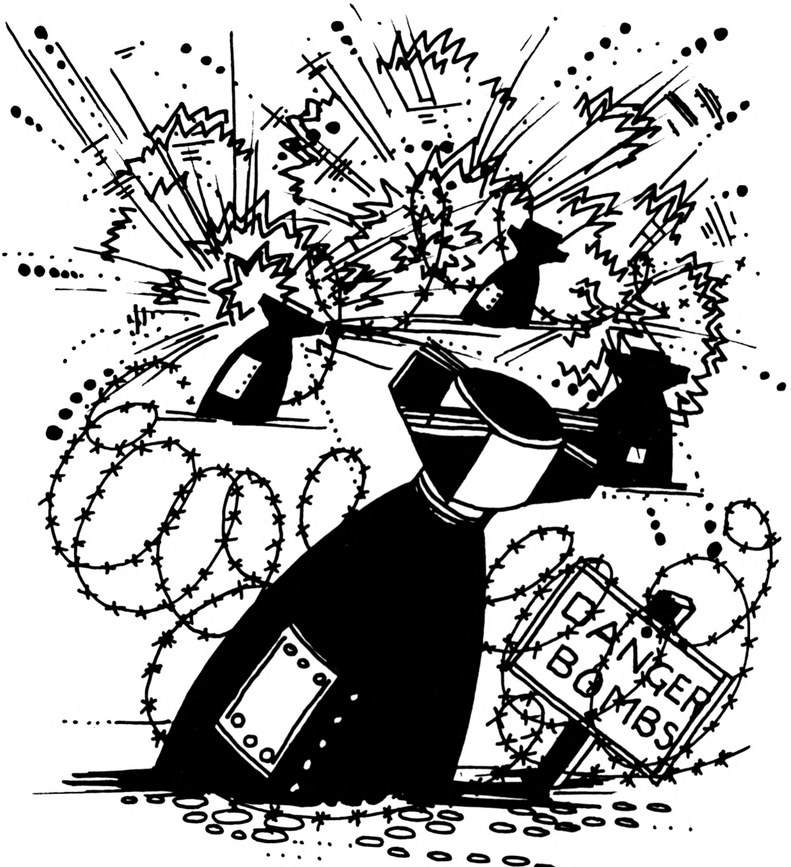
TREASURE CHEST



```
  10 RAND
  15 LET P=INT (RND*15)+10
  20 PRINT AT 0,0;"▌▌▌▌▌▌▌▌▌▌▌▌▌▌
▌▌▌▌▌▌▌▌▌▌▌"( TO P);AT 5,5;P;" C
HESTS"
  30 IF RND<.5 THEN GOTO 120
  40 PRINT AT 8,0;"
"
  50 PRINT AT 12,0;"HOW MANY CHE
STS WILL YOU REMOVE?"
  60 LET R=CODE INKEY$-28
  70 IF R<1 OR R>3 THEN GOTO 60
  80 PRINT AT 12,0;"
                         ";AT 9,0;"
            "
  90 LET P=P-R
 100 PRINT AT 5,5;P;" CHEST";"S"
 AND P<>1;" LEFT ";AT 8,0;"YOU R
EMOVED ";R;
 105 GOSUB 240
 110 IF P<1 THEN GOTO 220
 120 LET N=INT (P/4)*4
 130 IF P=N THEN LET R=INT (RND*
3)+1
 140 IF P<>N THEN LET R=P-N
 150 LET P=P-R
 160 PRINT AT 9,0;"I REMOVED ";R
 165 GOSUB 240
 170 PRINT AT 5,5;P;" CHEST";"S"
 AND P<>1;" LEFT "
 180 IF P<1 THEN GOTO 200
 190 GOTO 40
 200 PRINT AT 0,0;"£££ I WIN £££
"
 210 STOP
 220 PRINT AT 0,0;"£££ YOU WIN £
££"
 230 STOP
 240 FOR A=1 TO 10
 250 PRINT AT 0,P;"▌▌▌"( TO R);A
T 0,P;"   "( TO R)
 260 NEXT A
 270 RETURN
```

BOMB BLAST NIM

In this variation of Nim, also written by Dilwyn Jones, you lose if you are forced to take the last object. The objects are unexploded bombs which must be defused. The last bomb, however, is unstable and will explode and kill you if touched.



```
  10 RAND
  15 LET P=INT (RND*15)+10
  20 PRINT AT 0,0;"██████████████
████████████"( TO P);AT 5,5;P;" B
OMBS"
  30 IF RND<.5 THEN GOTO 120
  40 PRINT AT 8,0;"
"
  50 PRINT AT 12,0;"HOW MANY BOM
BS WILL YOU DEFUSE?"
  60 LET R=CODE INKEY$-28
  70 IF R<1 OR R>3 OR R>P THEN G
OTO 60
  80 PRINT AT 12,0;"
                  ";AT 9,0;"
        "
  90 LET P=P-R
 100 PRINT AT 5,5;P;" BOMB";"S"
AND P<>1;" LEFT ";AT 8,0;"YOU DE
FUSED ";R;
 105 GOSUB 240
 110 IF P<1 THEN GOTO 200
 120 LET N=INT (P/4)*4
 130 IF P=N+1 THEN LET R=INT (RN
D*3)+1
 140 IF P<>N+1 THEN LET R=P-(INT
 ((P-1)/4)*4)-1
 145 IF R>P THEN LET R=P
 150 LET P=P-R
 160 PRINT AT 9,0;"I DEFUSED ";R
 165 GOSUB 240
 170 PRINT AT 5,5;P;" BOMB";"S"
AND P<>1;" LEFT "
 180 IF P<1 THEN GOTO 220
 190 GOTO 40
 200 PRINT AT 0,0;" BANG  I WIN
 BANG ";AT 12,0;" BANG  I WIN  B
ANG "
 210 STOP
 220 PRINT AT 0,0;"BANG  YOU WIN
 BANG";AT 12,0;"BANG  YOU WIN
BANG"
 230 STOP
 240 FOR A=1 TO 10
 250 PRINT AT 0,P;"███"( TO A);A
T 0,P;"   "( TO A)
 260 NEXT A
 270 RETURN
```

POKER

This 16K program by Neil Streeter not only plays Poker but also displays the cards. Although this makes the program longer than it would otherwise be, the added enjoyment that comes from having the cards shown graphically compensates for the increased length. The display shows two hands, yours and the computer's, although you can't of course see the face of the computer's cards. You are then asked how much you want to bet.

The ranks of the hands are as follows:

LOWEST
Nothing — loses to everything above
One pair — odds one to two
Two pairs — odds one to 21
Three of a kind — odds one to 47
Straight — odds one to 255
Flush — odds one to 509
Full House — odds one to 694
Four of a Kind — odds one to 4,165
Straight Flush — one to 72,193
Royal Flush — one to 649,740
HIGHEST

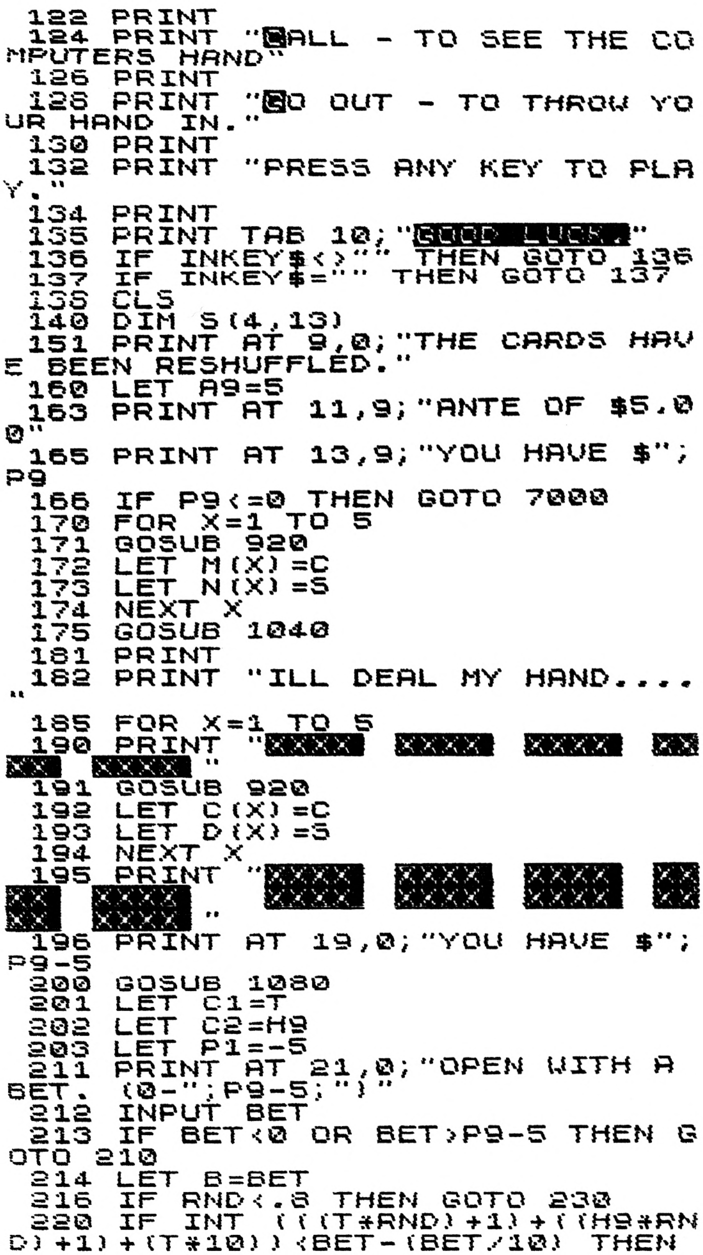
After your initial bet you may change up to three cards and try for a better hand. The computer can do the same. The computer will then reveal its cards, and compare the two hands. If you lose all of your money you may borrow from the computer and try to win it back. The program can be saved by starting the recorder and simply pressing RUN and ENTER. This means the program will start automatically on loading. Delete line 1 if you don't want this option.

Lines 1 to 105 set up the variables; 110 to 138 introduce the game; 140 to 592 hold the main game routine, while

597 to 638 is the end of game routine. Lines 640 to 793 analyse the computer's hand and lines 800 to 911 are the hand type subroutines. Lines 940 to 1002 print the card types, 1040 to 6120 print the screen routines, and the lines from 7000 to the end are the 'borrow money' routine. Line 190 is four inverse Xs, a graphic 7 and a space ten times.

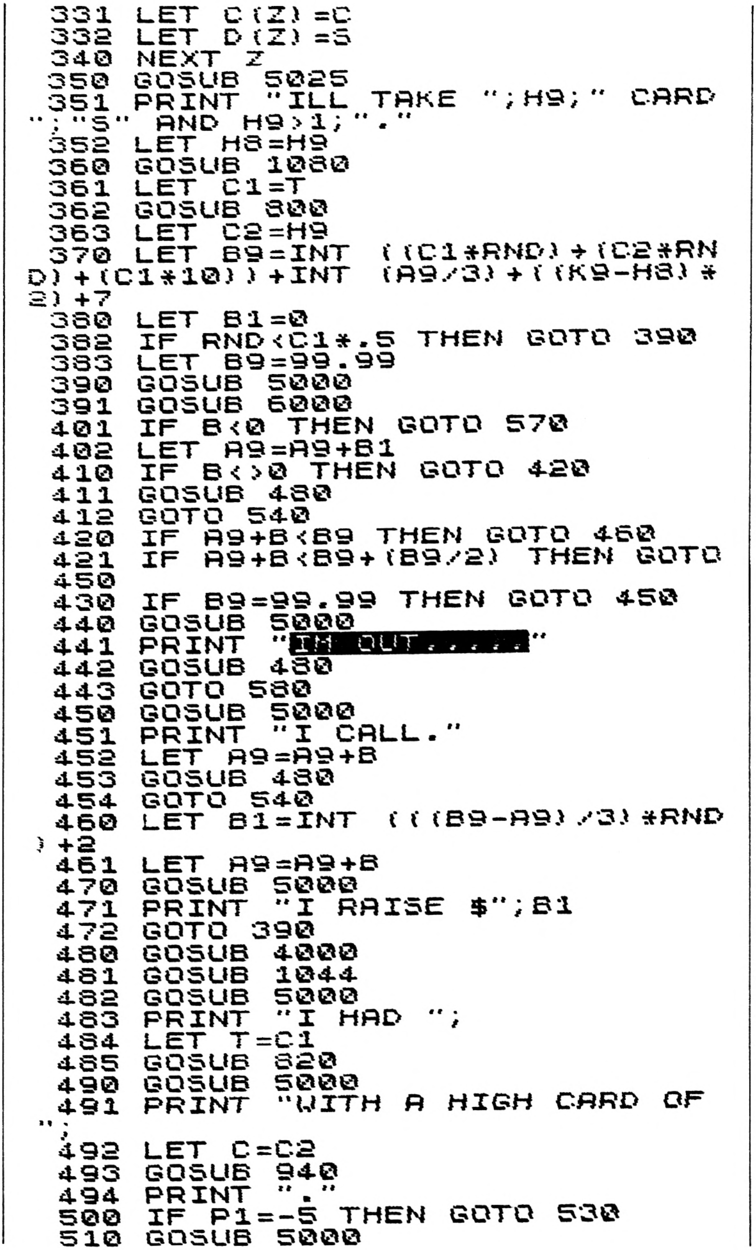
```
   1 SAVE "POKER"
   2 LET S$="
                                "
   5 LET BORROW=0
  10 RAND
  12 DIM A$(9,5)
  14 DIM T$(11,5)
  16 LET A$(1)="     "
  18 LET A$(2)="     "
  20 LET A$(3)="     "
  22 LET A$(4)="     "
  24 LET A$(5)="     "
  26 LET A$(6)="     "
  28 LET A$(7)="     "
  30 LET A$(8)="     "
  32 LET A$(9)="     "
```

```
  34 LET T$(1)="99992"
  36 LET T$(2)="68682"
  38 LET T$(3)="76872"
  40 LET T$(4)="48842"
  42 LET T$(5)="46842"
  44 LET T$(6)="45842"
  46 LET T$(7)="35842"
  48 LET T$(8)="44442"
  50 LET T$(9)="43442"
  52 LET T$(10)="34342"
  54 LET T$(11)="86882"
 100 DIM T(13)
 101 DIM K(4)
 102 DIM M(5)
 103 DIM N(5)
 104 DIM C(5)
 105 DIM D(5)
 110 PRINT TAB 8;"<< POKER GAME.
>>"
 111 LET P9=100
 112 PRINT
 114 PRINT "THIS IS THE GAME OF
POKER IN     WHICH YOU PLAY AGAIN
ST ME."
 116 PRINT "THERE IS AN ANTE BET
 OF $5.00 INEACH ROUND AND YOU S
TART WITH    $100. IN EACH ROUND
YOU MAY:-"
 118 PRINT
 120 PRINT "RAISE - TO RAISE BY
NOT MORE                THAN YOUR IN
ITIAL BET."
```

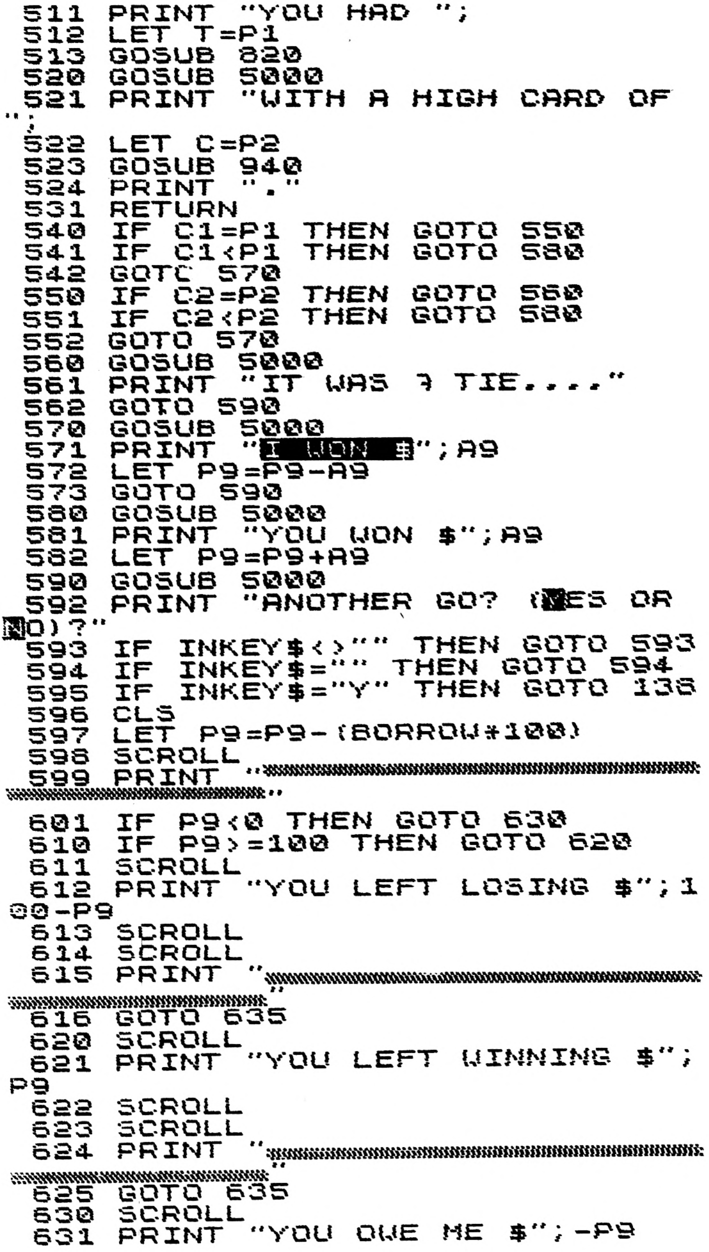


```
 122 PRINT
 124 PRINT "CALL - TO SEE THE CO
MPUTERS HAND"
 126 PRINT
 128 PRINT "GO OUT - TO THROW YO
UR HAND IN."
 130 PRINT
 132 PRINT "PRESS ANY KEY TO PLA
Y."
 134 PRINT
 135 PRINT TAB 10;"GOOD LUCK."
 136 IF INKEY$<>"" THEN GOTO 136
 137 IF INKEY$="" THEN GOTO 137
 138 CLS
 140 DIM S(4,13)
 151 PRINT AT 9,0;"THE CARDS HAV
E BEEN RESHUFFLED."
 160 LET A9=5
 163 PRINT AT 11,9;"ANTE OF $5.0
0"
 165 PRINT AT 13,9;"YOU HAVE $";
P9
 166 IF P9<=0 THEN GOTO 7000
 170 FOR X=1 TO 5
 171 GOSUB 920
 172 LET M(X)=C
 173 LET N(X)=S
 174 NEXT X
 175 GOSUB 1040
 181 PRINT
 182 PRINT "ILL DEAL MY HAND....
"
 185 FOR X=1 TO 5
 190 PRINT "▒▒▒▒▒  ▒▒▒▒▒  ▒▒▒▒▒  ▒▒
▒▒▒  ▒▒▒▒▒ "
 191 GOSUB 920
 192 LET C(X)=C
 193 LET D(X)=S
 194 NEXT X
 195 PRINT "▒▒▒▒▒  ▒▒▒▒▒  ▒▒▒▒▒  ▒▒
▒▒▒  ▒▒▒▒▒ "
 196 PRINT AT 19,0;"YOU HAVE $";
P9-5
 200 GOSUB 1080
 201 LET C1=T
 202 LET C2=H9
 203 LET P1=-5
 211 PRINT AT 21,0;"OPEN WITH A
BET. (0-";P9-5;")"
 212 INPUT BET
 213 IF BET<0 OR BET>P9-5 THEN G
OTO 210
 214 LET B=BET
 216 IF RND<.8 THEN GOTO 230
 220 IF INT (((T*RND)+1)+((H9*RN
D)+1)+(T*10))<BET-(BET/10) THEN
```

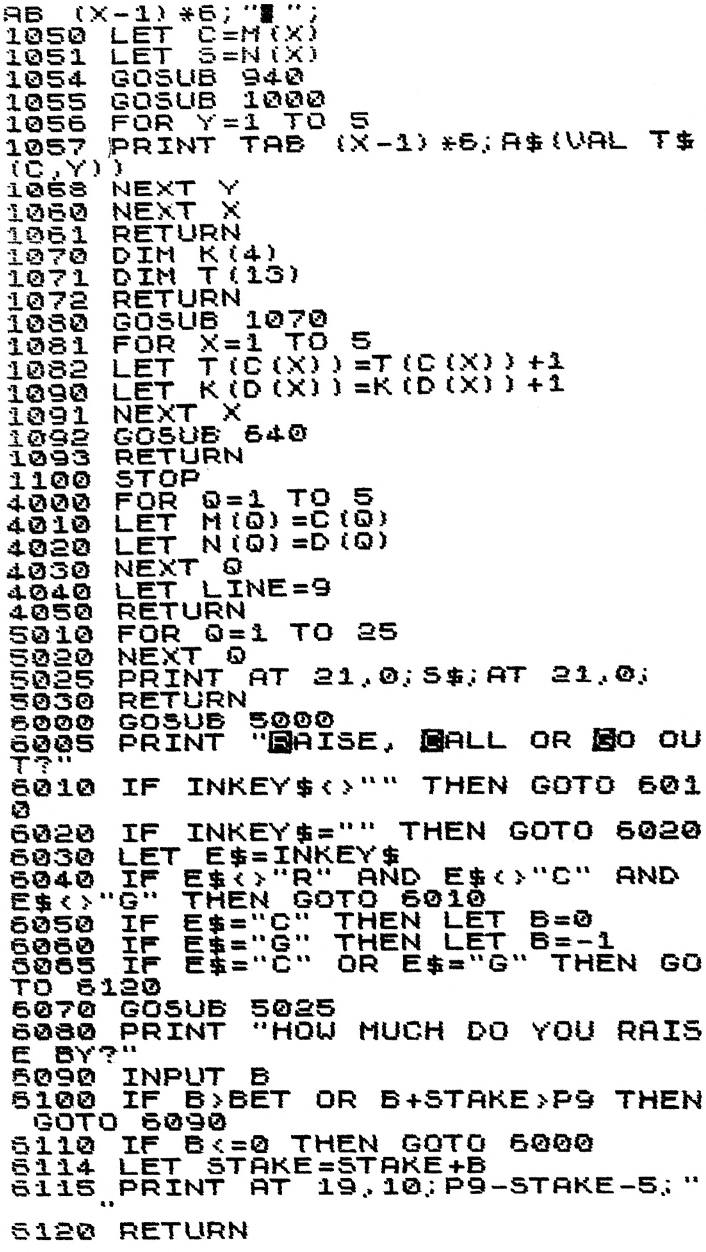
```
GOTO 440
 230 LET A9=A9+B
 231 GOSUB 5025
 232 PRINT "ILL ACCEPT."
 233 PRINT AT 19,10;P9-5-BET;"
 "
 234 LET STAKE=BET
 240 GOSUB 5000
 241 PRINT "HOW MANY CARDS TO RE
PLACE?"
 242 IF INKEY$<>"" THEN GOTO 242
 243 IF INKEY$="" THEN GOTO 243
 244 LET E$=INKEY$
 245 IF E$<"0" OR E$>"3" THEN GO
TO 242
 246 LET K9=VAL E$
 250 FOR X=1 TO K9
 260 GOSUB 5025
 261 PRINT "CARD NUMBER?" AND (X
/2=INT (X/2));"CARD NUMBER?" AND
 (X/2<>INT (X/2));" ";
 262 IF INKEY$<>"" THEN GOTO 262
 263 IF INKEY$="" THEN GOTO 263
 264 LET E$=INKEY$
 265 IF E$<"0" OR E$>"9" THEN GO
TO 262
 266 LET T9=VAL E$
 267 PRINT T9
 268 GOSUB 920
 270 IF T9<6 THEN GOTO 280
 271 GOSUB 5000
 272 PRINT "ENTER CARD NO. (1-5
ONLY)."
 273 GOTO 260
 280 LET M(T9)=C
 281 LET N(T9)=S
 282 NEXT X
 283 GOSUB 1044
 290 GOSUB 1070
 291 FOR X=1 TO 5
 292 LET T(M(X))=T(M(X))+1
 293 LET K(N(X))=K(N(X))+1
 294 NEXT X
 295 GOSUB 5025
 296 PRINT "I AM THINKING......"
 301 GOSUB 640
 302 LET P1=T
 303 GOSUB 800
 304 LET P2=H9
 311 GOSUB 1080
 312 LET H9=0
 313 IF T>3 THEN GOTO 350
 320 FOR Z=1 TO 5
 321 IF H9=3 THEN GOTO 340
 322 IF T(C(Z))<>1 THEN GOTO 340
 323 LET H9=H9+1
 330 GOSUB 920
```



```
 331 LET C(Z)=C
 332 LET D(Z)=S
 340 NEXT Z
 350 GOSUB 5025
 351 PRINT "ILL TAKE ";H9;" CARD
";"S" AND H9>1;"."
 352 LET H8=H9
 360 GOSUB 1080
 361 LET C1=T
 362 GOSUB 800
 363 LET C2=H9
 370 LET B9=INT ((C1*RND)+(C2*RN
D)+(C1*10))+INT (A9/3)+((K9-H8)*
2)+7
 380 LET B1=0
 382 IF RND<C1*.5 THEN GOTO 390
 383 LET B9=99.99
 390 GOSUB 5000
 391 GOSUB 6000
 401 IF B<0 THEN GOTO 570
 402 LET A9=A9+B1
 410 IF B<>0 THEN GOTO 420
 411 GOSUB 480
 412 GOTO 540
 420 IF A9+B<B9 THEN GOTO 460
 421 IF A9+B<B9+(B9/2) THEN GOTO
 450
 430 IF B9=99.99 THEN GOTO 450
 440 GOSUB 5000
 441 PRINT "IM OUT....."
 442 GOSUB 480
 443 GOTO 580
 450 GOSUB 5000
 451 PRINT "I CALL."
 452 LET A9=A9+B
 453 GOSUB 480
 454 GOTO 540
 460 LET B1=INT (((B9-A9)/3)*RND
)+2
 461 LET A9=A9+B
 470 GOSUB 5000
 471 PRINT "I RAISE $";B1
 472 GOTO 390
 480 GOSUB 4000
 481 GOSUB 1044
 482 GOSUB 5000
 483 PRINT "I HAD ";
 484 LET T=C1
 485 GOSUB 820
 490 GOSUB 5000
 491 PRINT "WITH A HIGH CARD OF
";
 492 LET C=C2
 493 GOSUB 940
 494 PRINT "."
 500 IF P1=-5 THEN GOTO 530
 510 GOSUB 5000
```



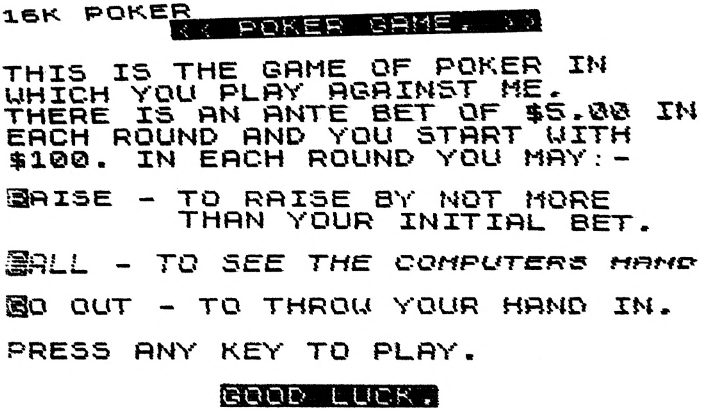
```
 511 PRINT "YOU HAD ";
 512 LET T=P1
 513 GOSUB 820
 520 GOSUB 5000
 521 PRINT "WITH A HIGH CARD OF
";
 522 LET C=P2
 523 GOSUB 940
 524 PRINT "."
 531 RETURN
 540 IF C1=P1 THEN GOTO 550
 541 IF C1<P1 THEN GOTO 580
 542 GOTC 570
 550 IF C2=P2 THEN GOTO 560
 551 IF C2<P2 THEN GOTO 580
 552 GOTO 570
 560 GOSUB 5000
 561 PRINT "IT WAS A TIE...."
 562 GOTO 590
 570 GOSUB 5000
 571 PRINT "I WON $";A9
 572 LET P9=P9-A9
 573 GOTO 590
 580 GOSUB 5000
 581 PRINT "YOU WON $";A9
 582 LET P9=P9+A9
 590 GOSUB 5000
 592 PRINT "ANOTHER GO? (YES OR
NO)?"
 593 IF INKEY$<>"" THEN GOTO 593
 594 IF INKEY$="" THEN GOTO 594
 595 IF INKEY$="Y" THEN GOTO 138
 596 CLS
 597 LET P9=P9-(BORROW*100)
 598 SCROLL
 599 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒"
 601 IF P9<0 THEN GOTO 630
 610 IF P9>=100 THEN GOTO 620
 611 SCROLL
 612 PRINT "YOU LEFT LOSING $";1
00-P9
 613 SCROLL
 614 SCROLL
 615 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒"
 616 GOTO 635
 620 SCROLL
 621 PRINT "YOU LEFT WINNING $";
P9
 622 SCROLL
 623 SCROLL
 624 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒"
 625 GOTO 635
 630 SCROLL
 631 PRINT "YOU OWE ME $";-P9
 632 SCROLL
 633 SCROLL
 634 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
 635 SCROLL
 636 SCROLL
 637 SCROLL
 638 STOP
 640 LET T=0
 641 FOR I=1 TO 4
 642 IF K(I)<>5 THEN GOTO 650
 643 LET T=5
 650 NEXT I
 651 LET I=2
 652 LET H9=0
 660 LET I=I-1
 661 IF I<>0 THEN GOTO 670
 662 LET I=13
 670 IF T(I)<1 THEN GOTO 660
 671 LET H9=I
 672 IF I<>1 THEN GOTO 680
 673 LET I=14
 680 LET Z=I-4
 690 LET I=I-1
 691 IF T(I)<>1 THEN GOTO 720
 700 IF Z<>1 THEN GOTO 690
 710 LET T=T+4
 711 IF Z<>10 THEN GOTO 720
 712 IF T=4 THEN GOTO 170
 713 LET T=T+1
 720 IF I=13 THEN GOTO 660
 721 IF T<>5 THEN GOTO 730
 722 IF T(1)<>1 THEN GOTO 730
 723 LET H9=1
 730 IF T=0 THEN GOTO 740
 731 RETURN
 740 FOR I=1 TO 13
 741 IF T(I)<>4 THEN GOTO 750
 742 LET T=7
 750 IF T(I)<>3 THEN GOTO 760
 751 LET T=T+5
 752 LET H9=I
 760 IF T(I)<>2 THEN GOTO 790
 761 IF T=5 THEN GOTO 780
 762 IF T<>0 THEN GOTO 770
 763 LET H9=0
 770 IF H9=1 THEN GOTO 780
 771 IF H9>I THEN GOTO 780
 772 LET H9=I
 780 LET T=T+1
 790 NEXT I
 791 IF T<>5 THEN GOTO 810
 792 LET T=3
 793 RETURN
 800 IF H9<>1 THEN GOTO 810
 801 LET H9=14
 810 RETURN
 820 IF T<>0 THEN GOTO 830
 821 PRINT "NOTHING"
 822 RETURN
 830 IF T<>1 THEN GOTO 840
 831 PRINT "ONE PAIR"
 832 RETURN
 840 IF T<>2 THEN GOTO 850
 841 PRINT "TWO PAIRS"
 842 RETURN
 850 IF T<>3 THEN GOTO 860
 851 PRINT "THREE OF A KIND"
 852 RETURN
 860 IF T<>4 THEN GOTO 870
 861 PRINT "A STRAIGHT"
 862 RETURN
 870 IF T<>5 THEN GOTO 880
 871 PRINT "A FLUSH"
 872 RETURN
 880 IF T<>6 THEN GOTO 890
 881 PRINT "A FULL HOUSE"
 882 RETURN
 890 IF T<>7 THEN GOTO 900
 891 PRINT "FOUR OF A KIND"
 892 RETURN
 900 IF T<>9 THEN GOTO 910
 901 PRINT "A STRAIGHT FLUSH"
 902 RETURN
 910 PRINT "A ROYAL FLUSH..."
 911 RETURN
 920 LET S=INT (RND*4+1)
 921 LET C=INT (RND*13+1)
 930 IF S(S,C)=1 THEN GOTO 920
 931 LET S(S,C)=1
 932 RETURN
 940 PRINT "A" AND (C=1 OR C=14)
;"T" AND C=10;"J" AND C=11;"Q" A
ND C=12;"K" AND C=13;
 941 IF C>1 AND C<10 THEN PRINT
C;
 942 IF C=1 THEN LET C=14
 943 IF C>10 AND C<14 THEN LET C
=1
 944 IF C=14 THEN LET C=11
 952 RETURN
1000 PRINT "H" AND S=1;"S" AND S
=2;"D" AND S=3;"C" AND S=4;
1001 PRINT " █"
1002 RETURN
1040 CLS
1041 LET LINE=0
1042 PRINT TAB 7;"HERE IS YOUR H
AND."
1043 PRINT "  1     2     3
4     5"
1044 FOR X=1 TO 5
1045 PRINT AT 2+LINE,0;
1046 PRINT TAB (X-1)*6;"▛▀▀▀▀▜ ";T
```



```
AB  (X-1)*6;"█";
1050 LET C=M(X)
1051 LET S=N(X)
1054 GOSUB 940
1055 GOSUB 1000
1056 FOR Y=1 TO 5
1057 PRINT TAB (X-1)*6;A$(VAL T$
(C,Y))
1068 NEXT Y
1060 NEXT X
1061 RETURN
1070 DIM K(4)
1071 DIM T(13)
1072 RETURN
1080 GOSUB 1070
1081 FOR X=1 TO 5
1082 LET T(C(X))=T(C(X))+1
1090 LET K(D(X))=K(D(X))+1
1091 NEXT X
1092 GOSUB 640
1093 RETURN
1100 STOP
4000 FOR Q=1 TO 5
4010 LET M(Q)=C(Q)
4020 LET N(Q)=D(Q)
4030 NEXT Q
4040 LET LINE=9
4050 RETURN
5010 FOR Q=1 TO 25
5020 NEXT Q
5025 PRINT AT 21,0;S$;AT 21,0;
5030 RETURN
6000 GOSUB 5000
6005 PRINT "RAISE, CALL OR GO OU
T?"
6010 IF INKEY$<>"" THEN GOTO 601
0
6020 IF INKEY$="" THEN GOTO 6020
6030 LET E$=INKEY$
6040 IF E$<>"R" AND E$<>"C" AND
E$<>"G" THEN GOTO 6010
6050 IF E$="C" THEN LET B=0
6060 IF E$="G" THEN LET B=-1
6065 IF E$="C" OR E$="G" THEN GO
TO 6120
6070 GOSUB 5025
6080 PRINT "HOW MUCH DO YOU RAIS
E BY?"
6090 INPUT B
6100 IF B>BET OR B+STAKE>P9 THEN
 GOTO 6090
6110 IF B<=0 THEN GOTO 6000
6114 LET STAKE=STAKE+B
6115 PRINT AT 19,10;P9-STAKE-5;"
  "
6120 RETURN
```

```
7000 PRINT
7010 PRINT "YOU HAVE NO MONEY DO
 YOU WISH TOBORROW $100? (YES OR
 NO)?"
7020 IF INKEY$<>"" THEN GOTO 702
0
7030 IF INKEY$="" THEN GOTO 7030
7040 LET E$=INKEY$
7050 IF E$<>"Y" THEN GOTO 596
7060 LET P9=100
7065 LET BORROW=BORROW+1
7070 GOTO 138
```

16K POKER



<< POKER GAME. >>

THIS IS THE GAME OF POKER IN
WHICH YOU PLAY AGAINST ME.
THERE IS AN ANTE BET OF $5.00 IN
EACH ROUND AND YOU START WITH
$100. IN EACH ROUND YOU MAY:-

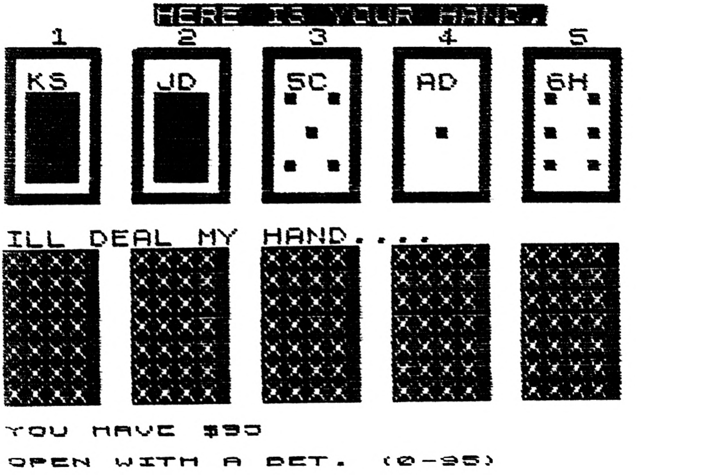
RAISE - TO RAISE BY NOT MORE
        THAN YOUR INITIAL BET.

CALL - TO SEE THE COMPUTERS HAND

GO OUT - TO THROW YOUR HAND IN.

PRESS ANY KEY TO PLAY.

GOOD LUCK.



HERE IS YOUR HAND.

ILL DEAL MY HAND....

YOU HAVE $95

OPEN WITH A BET. (0-95)

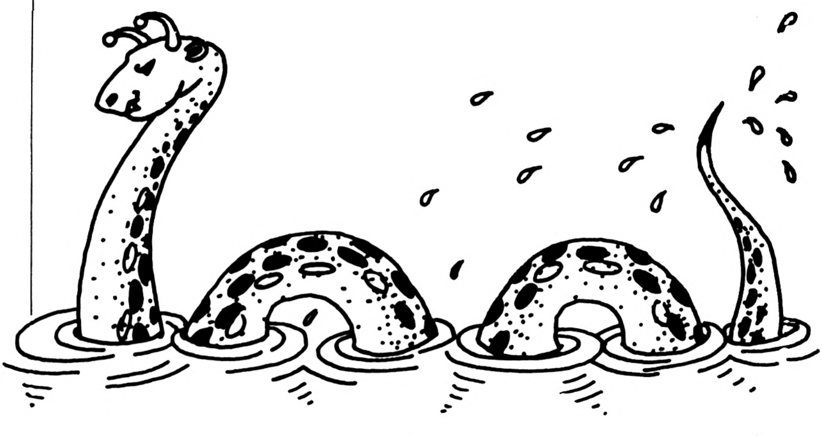
NESSIE

In this game (written by Dilwyn Jones and using just 1K) the Loch Ness Monster is hiding behind a row of rocks on the shore of the Loch, and you have to destroy it before it can eat you. Before you can do this, however, you must find out where the Monster is by pressing the key corresponding to the number on the rock. The rock will be destroyed when you do this, and if the Monster is behind the rock it will emerge into view after the dust has settled. If you do locate Nessie, you must press any of the *letter* keys to destroy it, before it destroys you.

Line 10 generates a number from one to nine, and turns it into a string character with the STR$ function, so that it can be tested later against the key being pressed on the keyboard. The numbers in inverse video are the rocks (you may have expected high resolution graphics). Line 30 scans the keyboard, and line 40 prevents the program from continuing until one of the keys 1 to 9 has been pressed. Line 50 prints a space where the rock has been destroyed. Lines 60 to 80 produce a random delay to represent the time taken for the dust to settle, and you must not press a letter key during this delay time.

If Nessie is not behind the rock you have destroyed, line 90 makes you try again to find it. However, if the Monster is behind the rock, then the program continues with line 100 which brings it into view. The loop in lines 110 to 130 gives the player a short time in which to press a letter key, and if this is not done within that limit then Nessie attacks. Enter any word beginning with 'Y' (YUP, YES, YEAH, or whatever) at the end if you want another game.

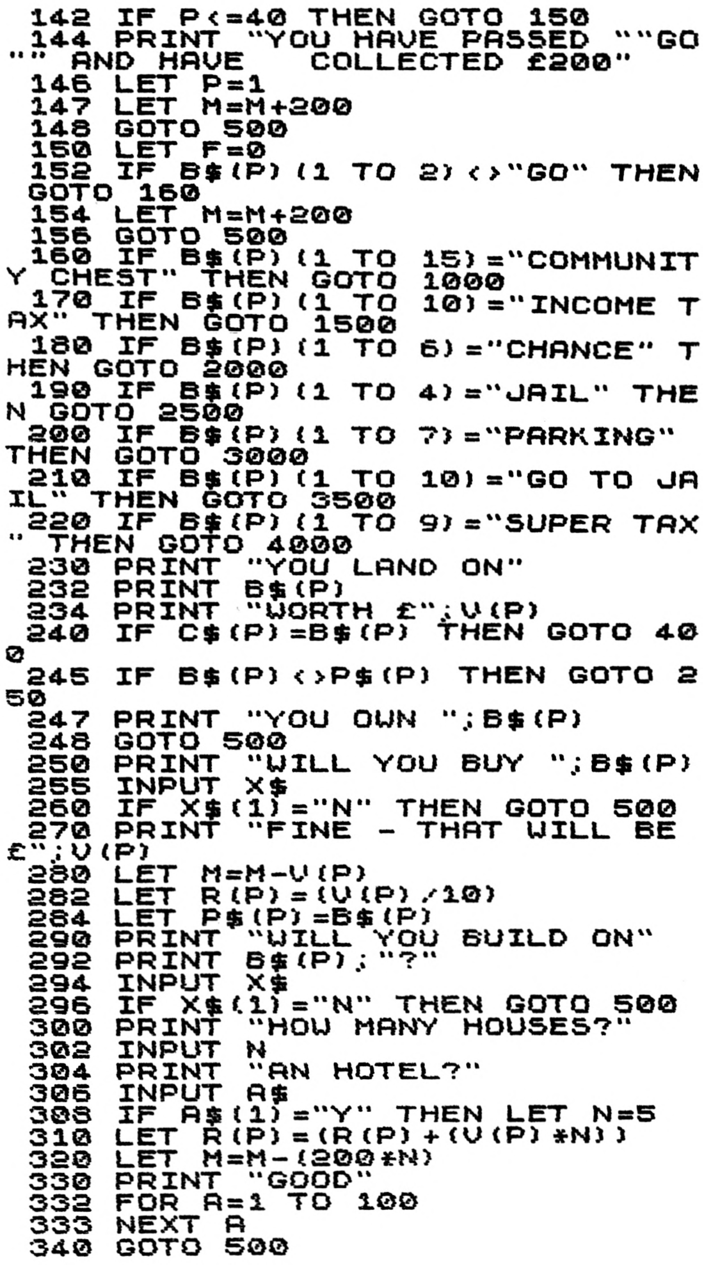
```
   1 RAND
  10 LET U$=STR$ (INT (RND*9)+1)
  20 PRINT AT 3,3;"▒  ▒  ▒  ▒  ▒
▒  ▒  ▒  ▒";AT 8,14;"▒▒▒▒";AT 9
,14;"▒▒▒"
  30 LET A$=INKEY$
  40 IF A$<"1" OR A$>"9" THEN GO
TO 30
  50 PRINT AT 3,VAL A$*3;" "
  60 FOR I=1 TO RND*50+10
  70 IF INKEY$>"9" THEN GOTO 210
  80 NEXT I
  90 IF A$<>U$ THEN GOTO 30
 100 PRINT AT 3,VAL U$*3;"*"
 110 FOR I=1 TO 8
 120 IF INKEY$>"9" THEN GOTO 190
 130 NEXT I
 140 PRINT AT 8,14;"▒▒▒▒▒▒▒▒▒";T
AB 14;"    ","OOPS, WHO""S A BIT
SLOW THEN?"
 145 PRINT AT 12,0;"ANOTHER GO?"
 150 INPUT A$
 160 IF CODE A$<>CODE "Y" THEN S
TOP
 170 CLS
 180 RUN
 190 PRINT AT 3,(VAL U$*3-2);"GO
T IT"
 200 GOTO 145
 210 PRINT AT 0,5;"TRIGGER HAPPY
 PERSON"
 220 GOTO 145
```



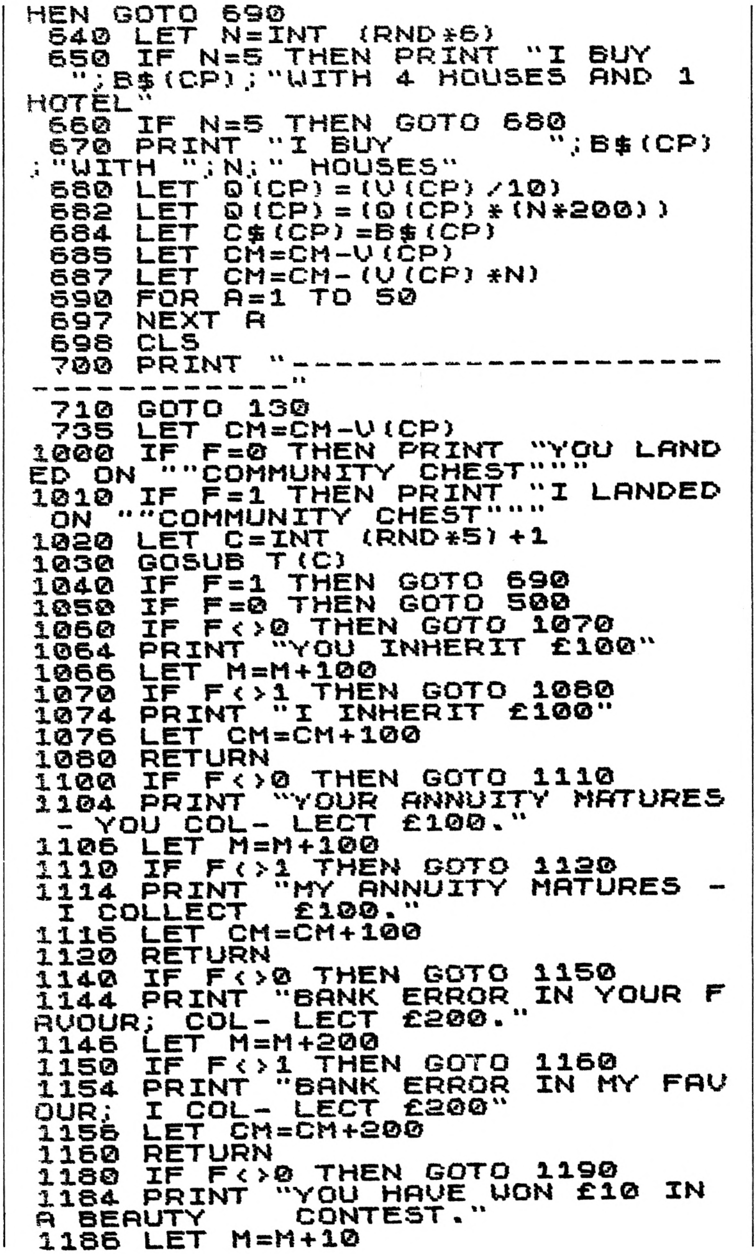
POLOMONY

Here is your chance to make a million without any interference or competition from other human beings. Of course, you'll have to defeat the cunning ZX81 in your attempt to control the streets and properties of London. This program follows the rules of the well-known board game Monopoly, and you may well find your enjoyment of the game is enhanced if you set up a Monopoly board by the television while you are playing. Polomony was written by Chris Callender, and adapted by Clive Gifford.

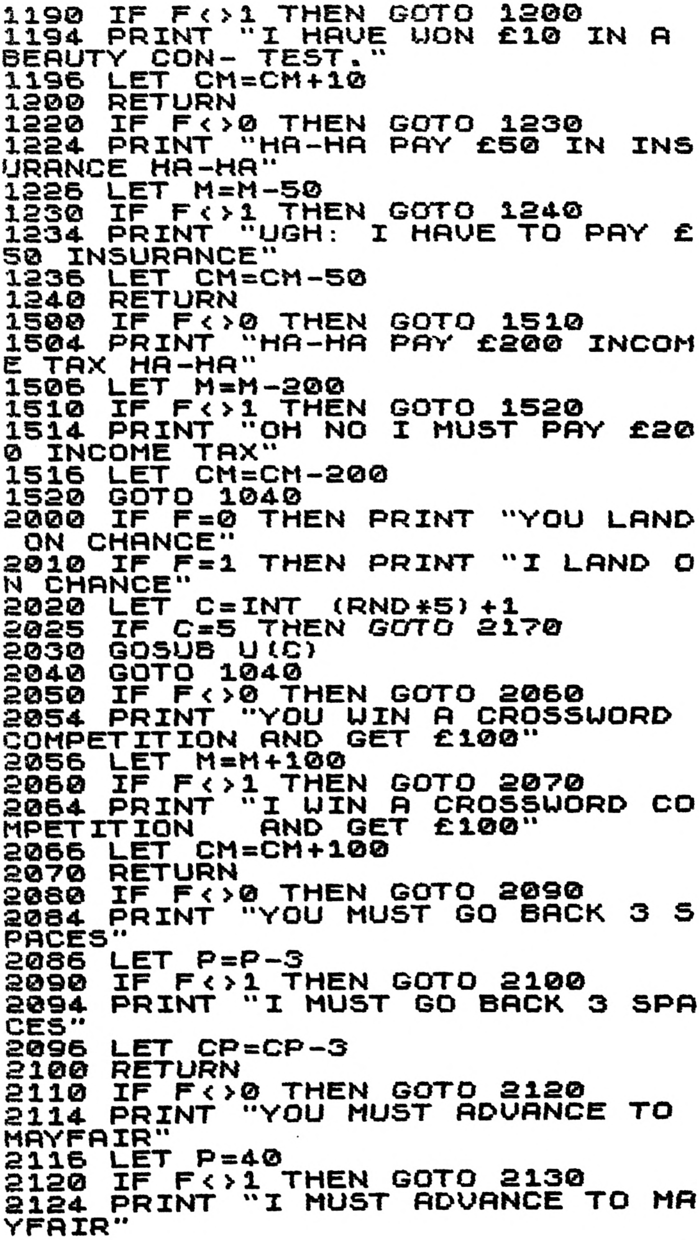
```
   1 SLOW
   2 REM POLOMONY
   3 PRINT "        ▓▓▓▓▓▓▓▓▓▓▓
▓▓▓"
   4 PRINT "        ▓▓▓▓▓▓▓▓▓▓▓
▓▓▓"
   6 PRINT "        ▓▓ P
 ▓▓"
   8 PRINT "        ▓▓  O ▓▓▓
 ▓▓"
  10 PRINT "        ▓▓   L▓▓▓
 ▓▓"
  12 PRINT "        ▓▓    O
 ▓▓"
  14 PRINT "        ▓▓     M
 ▓▓"
  16 PRINT "        ▓▓  ▓▓▓ O
 ▓▓"
  18 PRINT "        ▓▓  ▓▓▓  N
 ▓▓"
  20 PRINT "        ▓▓        Y
 ▓▓"
  22 PRINT "        ▓▓▓▓▓▓▓▓▓▓▓
▓▓▓"
  24 PRINT "        ▓▓▓▓▓▓▓▓▓▓▓
▓▓▓"
  26 PRINT AT 14,0;"PRESS ANY KE
Y TO CONTINUE"
  28 IF INKEY$="" THEN GOTO 28
  30 CLS
  60 DIM B$(40,21)
  62 GOTO 9000
  64 DIM U(4)
  66 FOR F=1 TO 4
  68 LET U(F)=F*30+2020
  70 NEXT F
  72 DIM T(5)
  74 FOR F=1 TO 5
  76 LET T(F)=F*40+1020
  78 NEXT F
  80 DIM P$(40,21)
  82 DIM C$(40,21)
  84 DIM V(40)
  86 DIM R(40)
  88 DIM Q(40)
  90 GOTO 9082
 100 LET M=9000
 110 LET CM=9000
 115 LET P=1
 120 LET CP=1
 130 PRINT "PRESS ""NEWLINE"" TO
ROLL THE DICE"
 132 INPUT A$
 134 LET D=INT (RND*6)+1
 136 PRINT "DICE =";D
 140 LET P=P+D
```



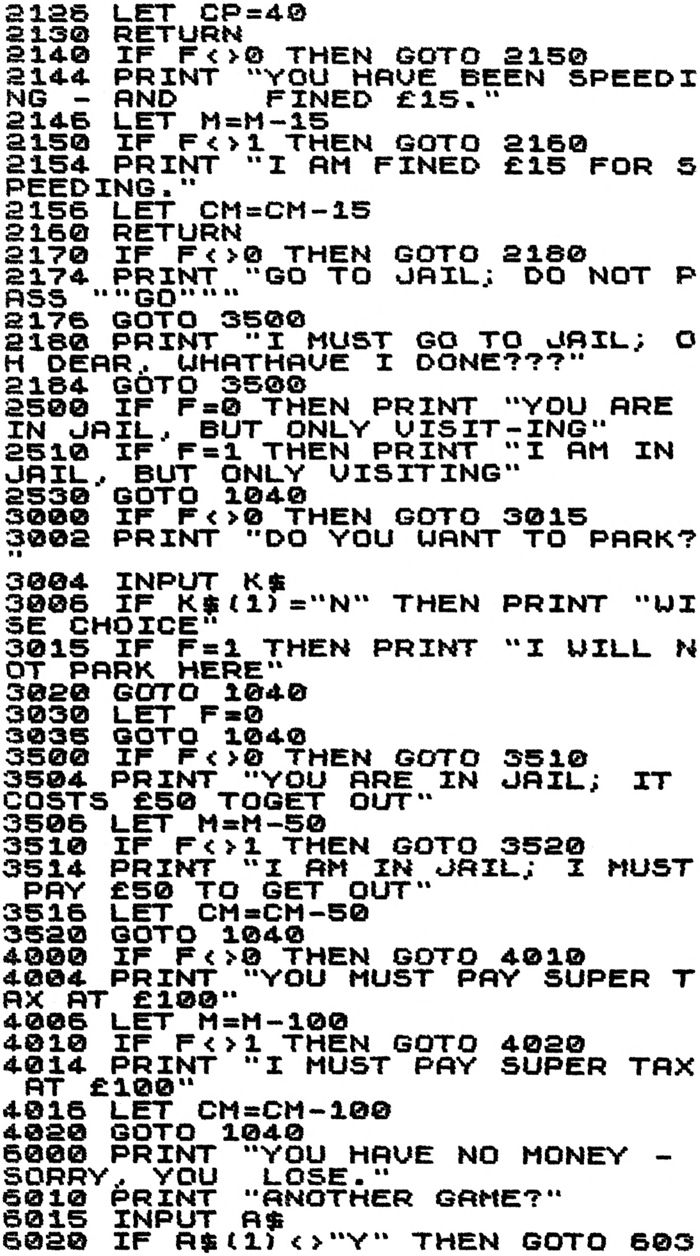
```
 142 IF P<=40 THEN GOTO 150
 144 PRINT "YOU HAVE PASSED ""GO
"" AND HAVE   COLLECTED £200"
 146 LET P=1
 147 LET M=M+200
 148 GOTO 500
 150 LET F=0
 152 IF B$(P)(1 TO 2)<>"GO" THEN
 GOTO 160
 154 LET M=M+200
 156 GOTO 500
 160 IF B$(P)(1 TO 15)="COMMUNIT
Y CHEST" THEN GOTO 1000
 170 IF B$(P)(1 TO 10)="INCOME T
AX" THEN GOTO 1500
 180 IF B$(P)(1 TO 6)="CHANCE" T
HEN GOTO 2000
 190 IF B$(P)(1 TO 4)="JAIL" THE
N GOTO 2500
 200 IF B$(P)(1 TO 7)="PARKING"
THEN GOTO 3000
 210 IF B$(P)(1 TO 10)="GO TO JA
IL" THEN GOTO 3500
 220 IF B$(P)(1 TO 9)="SUPER TAX
" THEN GOTO 4000
 230 PRINT "YOU LAND ON"
 232 PRINT B$(P)
 234 PRINT "WORTH £";V(P)
 240 IF C$(P)=B$(P) THEN GOTO 40
0
 245 IF B$(P)<>P$(P) THEN GOTO 2
50
 247 PRINT "YOU OWN ";B$(P)
 248 GOTO 500
 250 PRINT "WILL YOU BUY ";B$(P)
 255 INPUT X$
 260 IF X$(1)="N" THEN GOTO 500
 270 PRINT "FINE - THAT WILL BE
£";V(P)
 280 LET M=M-V(P)
 282 LET R(P)=(V(P)/10)
 284 LET P$(P)=B$(P)
 290 PRINT "WILL YOU BUILD ON"
 292 PRINT B$(P);"?"
 294 INPUT X$
 296 IF X$(1)="N" THEN GOTO 500
 300 PRINT "HOW MANY HOUSES?"
 302 INPUT N
 304 PRINT "AN HOTEL?"
 306 INPUT A$
 308 IF A$(1)="Y" THEN LET N=5
 310 LET R(P)=(R(P)+(V(P)*N))
 320 LET M=M-(200*N)
 330 PRINT "GOOD"
 332 FOR A=1 TO 100
 333 NEXT A
 340 GOTO 500
 400 PRINT "YOU ARE TRESPASSING
ON MY PROP-"
 410 PRINT "ERTY - PAY £";Q(P)
 414 LET M=M-Q(P)
 416 LET CM=CM+Q(P)
 420 PRINT "THANK YOU VERY MUCH.
"
 500 PRINT "YOU NOW HAVE £";M
 502 PRINT "I HAVE £";CM
 503 FOR Y=1 TO 100
 504 NEXT Y
 505 CLS
 507 PRINT "--------------------
------------"
 510 IF M<0 THEN GOTO 6000
 515 IF CM<0 THEN GOTO 7000
 520 PRINT "MY SHOT"
 525 LET CD=INT (RND*6)+1
 530 LET CP=CP+CD
 535 PRINT "DICE=";CD
 540 IF CP<=40 THEN GOTO 550
 542 LET CP=1
 544 LET CM=CM+200
 545 PRINT "I HAVE PASSED GO"
 546 GOTO 690
 550 LET F=1
 560 IF B$(CP)(1 TO 15)="COMMUNI
TY CHEST" THEN GOTO 1000
 570 IF B$(CP)(1 TO 10)="INCOME
TAX" THEN GOTO 1500
 580 IF B$(CP)(1 TO 6)="CHANCE"
THEN GOTO 2000
 590 IF B$(CP)(1 TO 4)="JAIL" TH
EN GOTO 2500
 600 IF B$(CP)(1 TO 7)="PARKING"
 THEN GOTO 3000
 610 IF B$(CP)(1 TO 10)="GO TO J
AIL" THEN GOTO 3500
 615 IF B$(CP)(1 TO 9)="SUPER TA
X" THEN GOTO 4000
 618 IF P$(CP)<>B$(CP) THEN GOTO
 624
 619 LET CM=CM-R(CP)
 620 LET M=M+R(CP)
 621 PRINT "I TRESPASS ON"
 622 PRINT B$(CP)
 623 PRINT "AND PAY YOU £";R(CP)
 624 IF P$(CP)=B$(CP) THEN GOTO
690
 625 IF V(CP)=0 THEN GOTO 690
 526 IF B$(CP)<>C$(CP) THEN GOTO
 629
 627 PRINT "I LANDED ON ";B$(CP)
;"WHICH I OWN"
 628 GOTO 690
 629 PRINT "I LANDED ON ";B$(CP)
 630 IF CM/V(CP)<10 OR CM<1000 T
```



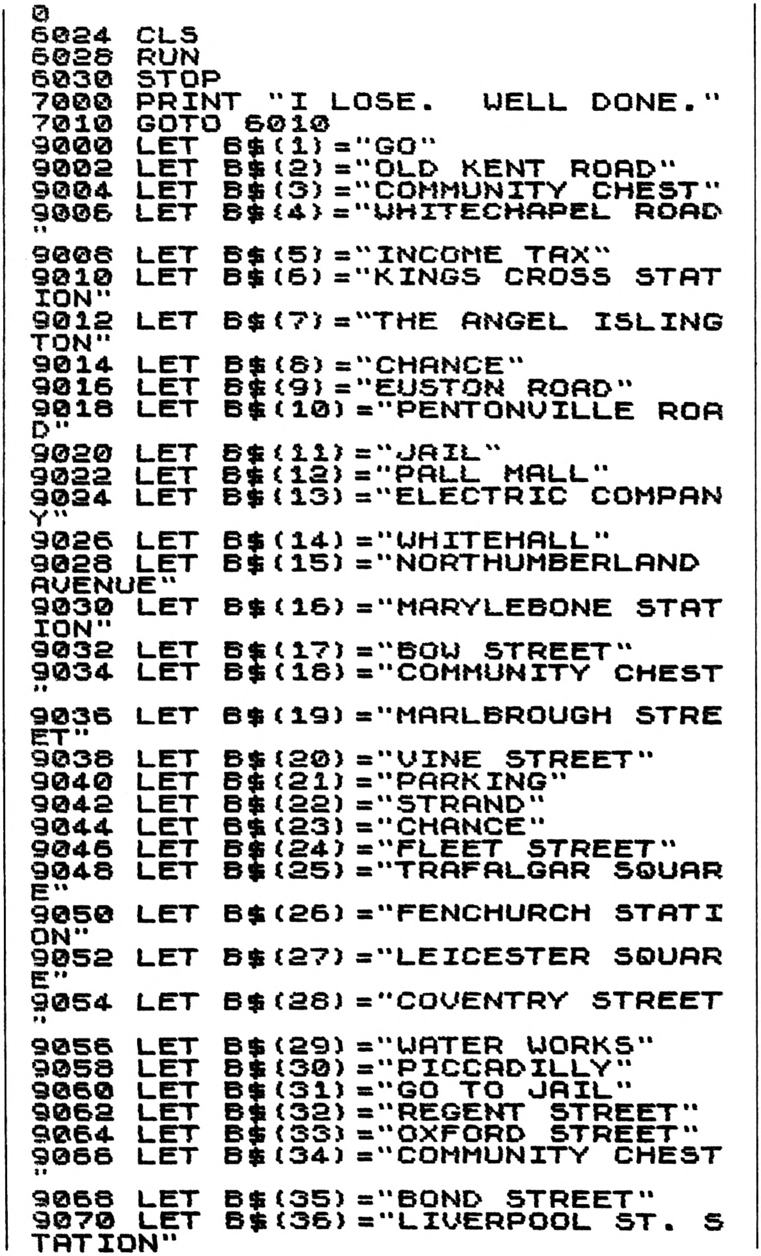
```
HEN GOTO 690
 640 LET N=INT (RND*6)
 650 IF N=5 THEN PRINT "I BUY
  ";B$(CP);"WITH 4 HOUSES AND 1
HOTEL"
 660 IF N=5 THEN GOTO 680
 670 PRINT "I BUY            ";B$(CP)
;"WITH ";N;" HOUSES"
 680 LET Q(CP)=(V(CP)/10)
 682 LET Q(CP)=(Q(CP)*(N*200))
 684 LET C$(CP)=B$(CP)
 685 LET CM=CM-V(CP)
 687 LET CM=CM-(V(CP)*N)
 690 FOR A=1 TO 50
 697 NEXT A
 698 CLS
 700 PRINT "--------------------
-------------"
 710 GOTO 130
 735 LET CM=CM-V(CP)
1000 IF F=0 THEN PRINT "YOU LAND
ED ON ""COMMUNITY CHEST"""
1010 IF F=1 THEN PRINT "I LANDED
 ON ""COMMUNITY CHEST"""
1020 LET C=INT (RND*5)+1
1030 GOSUB T(C)
1040 IF F=1 THEN GOTO 690
1050 IF F=0 THEN GOTO 500
1060 IF F<>0 THEN GOTO 1070
1064 PRINT "YOU INHERIT £100"
1066 LET M=M+100
1070 IF F<>1 THEN GOTO 1080
1074 PRINT "I INHERIT £100"
1076 LET CM=CM+100
1080 RETURN
1100 IF F<>0 THEN GOTO 1110
1104 PRINT "YOUR ANNUITY MATURES
 - YOU COL- LECT £100."
1106 LET M=M+100
1110 IF F<>1 THEN GOTO 1120
1114 PRINT "MY ANNUITY MATURES -
 I COLLECT  £100."
1116 LET CM=CM+100
1120 RETURN
1140 IF F<>0 THEN GOTO 1150
1144 PRINT "BANK ERROR IN YOUR F
AVOUR; COL- LECT £200."
1146 LET M=M+200
1150 IF F<>1 THEN GOTO 1160
1154 PRINT "BANK ERROR IN MY FAV
OUR; I COL- LECT £200"
1156 LET CM=CM+200
1160 RETURN
1180 IF F<>0 THEN GOTO 1190
1184 PRINT "YOU HAVE WON £10 IN
A BEAUTY     CONTEST."
1186 LET M=M+10
```



```
1190 IF F<>1 THEN GOTO 1200
1194 PRINT "I HAVE WON £10 IN A
BEAUTY CON- TEST."
1196 LET CM=CM+10
1200 RETURN
1220 IF F<>0 THEN GOTO 1230
1224 PRINT "HA-HA PAY £50 IN INS
URANCE HA-HA"
1226 LET M=M-50
1230 IF F<>1 THEN GOTO 1240
1234 PRINT "UGH: I HAVE TO PAY £
50 INSURANCE"
1236 LET CM=CM-50
1240 RETURN
1500 IF F<>0 THEN GOTO 1510
1504 PRINT "HA-HA PAY £200 INCOM
E TAX HA-HA"
1506 LET M=M-200
1510 IF F<>1 THEN GOTO 1520
1514 PRINT "OH NO I MUST PAY £20
0 INCOME TAX"
1516 LET CM=CM-200
1520 GOTO 1040
2000 IF F=0 THEN PRINT "YOU LAND
 ON CHANCE"
2010 IF F=1 THEN PRINT "I LAND O
N CHANCE"
2020 LET C=INT (RND*5)+1
2025 IF C=5 THEN GOTO 2170
2030 GOSUB U(C)
2040 GOTO 1040
2050 IF F<>0 THEN GOTO 2060
2054 PRINT "YOU WIN A CROSSWORD
COMPETITION AND GET £100"
2056 LET M=M+100
2060 IF F<>1 THEN GOTO 2070
2064 PRINT "I WIN A CROSSWORD CO
MPETITION   AND GET £100"
2066 LET CM=CM+100
2070 RETURN
2080 IF F<>0 THEN GOTO 2090
2084 PRINT "YOU MUST GO BACK 3 S
PACES"
2086 LET P=P-3
2090 IF F<>1 THEN GOTO 2100
2094 PRINT "I MUST GO BACK 3 SPA
CES"
2096 LET CP=CP-3
2100 RETURN
2110 IF F<>0 THEN GOTO 2120
2114 PRINT "YOU MUST ADVANCE TO
MAYFAIR"
2116 LET P=40
2120 IF F<>1 THEN GOTO 2130
2124 PRINT "I MUST ADVANCE TO MA
YFAIR"
```

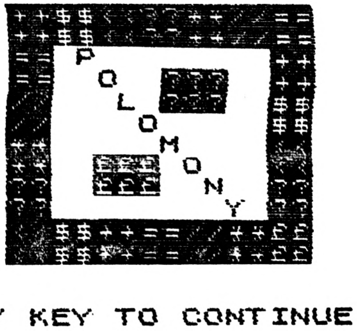


```
2126 LET CP=40
2130 RETURN
2140 IF F<>0 THEN GOTO 2150
2144 PRINT "YOU HAVE BEEN SPEEDI
NG - AND    FINED £15."
2146 LET M=M-15
2150 IF F<>1 THEN GOTO 2160
2154 PRINT "I AM FINED £15 FOR S
PEEDING."
2156 LET CM=CM-15
2160 RETURN
2170 IF F<>0 THEN GOTO 2180
2174 PRINT "GO TO JAIL; DO NOT P
ASS ""GO"""
2176 GOTO 3500
2180 PRINT "I MUST GO TO JAIL; O
H DEAR, WHATHAVE I DONE???"
2184 GOTO 3500
2500 IF F=0 THEN PRINT "YOU ARE
IN JAIL, BUT ONLY VISIT-ING"
2510 IF F=1 THEN PRINT "I AM IN
JAIL, BUT ONLY VISITING"
2530 GOTO 1040
3000 IF F<>0 THEN GOTO 3015
3002 PRINT "DO YOU WANT TO PARK?
"

3004 INPUT K$
3006 IF K$(1)="N" THEN PRINT "WI
SE CHOICE"
3015 IF F=1 THEN PRINT "I WILL N
OT PARK HERE"
3020 GOTO 1040
3030 LET F=0
3035 GOTO 1040
3500 IF F<>0 THEN GOTO 3510
3504 PRINT "YOU ARE IN JAIL; IT
COSTS £50 TOGET OUT"
3506 LET M=M-50
3510 IF F<>1 THEN GOTO 3520
3514 PRINT "I AM IN JAIL; I MUST
 PAY £50 TO GET OUT"
3516 LET CM=CM-50
3520 GOTO 1040
4000 IF F<>0 THEN GOTO 4010
4004 PRINT "YOU MUST PAY SUPER T
AX AT £100"
4006 LET M=M-100
4010 IF F<>1 THEN GOTO 4020
4014 PRINT "I MUST PAY SUPER TAX
 AT £100"
4016 LET CM=CM-100
4020 GOTO 1040
6000 PRINT "YOU HAVE NO MONEY -
SORRY, YOU  LOSE."
6010 PRINT "ANOTHER GAME?"
6015 INPUT A$
6020 IF A$(1)<>"Y" THEN GOTO 603
```

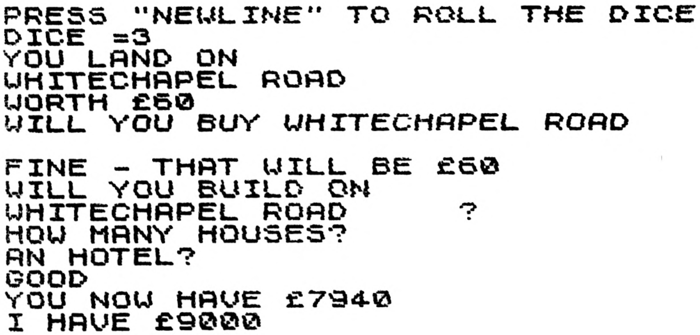


```
0
6024 CLS
6026 RUN
6030 STOP
7000 PRINT "I LOSE.  WELL DONE."
7010 GOTO 6010
9000 LET B$(1)="GO"
9002 LET B$(2)="OLD KENT ROAD"
9004 LET B$(3)="COMMUNITY CHEST"
9006 LET B$(4)="WHITECHAPEL ROAD
"
9008 LET B$(5)="INCOME TAX"
9010 LET B$(6)="KINGS CROSS STAT
ION"
9012 LET B$(7)="THE ANGEL ISLING
TON"
9014 LET B$(8)="CHANCE"
9016 LET B$(9)="EUSTON ROAD"
9018 LET B$(10)="PENTONVILLE ROA
D"
9020 LET B$(11)="JAIL"
9022 LET B$(12)="PALL MALL"
9024 LET B$(13)="ELECTRIC COMPAN
Y"
9026 LET B$(14)="WHITEHALL"
9028 LET B$(15)="NORTHUMBERLAND
AVENUE"
9030 LET B$(16)="MARYLEBONE STAT
ION"
9032 LET B$(17)="BOW STREET"
9034 LET B$(18)="COMMUNITY CHEST
"
9036 LET B$(19)="MARLBROUGH STRE
ET"
9038 LET B$(20)="VINE STREET"
9040 LET B$(21)="PARKING"
9042 LET B$(22)="STRAND"
9044 LET B$(23)="CHANCE"
9046 LET B$(24)="FLEET STREET"
9048 LET B$(25)="TRAFALGAR SQUAR
E"
9050 LET B$(26)="FENCHURCH STATI
ON"
9052 LET B$(27)="LEICESTER SQUAR
E"
9054 LET B$(28)="COVENTRY STREET
"
9056 LET B$(29)="WATER WORKS"
9058 LET B$(30)="PICCADILLY"
9060 LET B$(31)="GO TO JAIL"
9062 LET B$(32)="REGENT STREET"
9064 LET B$(33)="OXFORD STREET"
9066 LET B$(34)="COMMUNITY CHEST
"
9068 LET B$(35)="BOND STREET"
9070 LET B$(36)="LIVERPOOL ST. S
TATION"
```

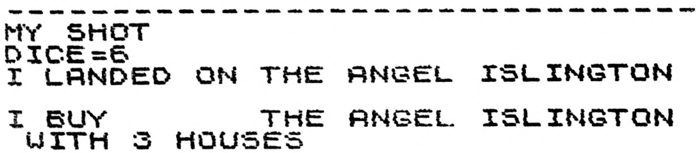
```
9072 LET B$(37)="CHANCE"
9074 LET B$(38)="PARK LANE"
9076 LET B$(39)="SUPER TAX"
9078 LET B$(40)="MAYFAIR"
9080 GOTO 64
9082 LET V(1)=0
9084 LET V(2)=60
9086 LET V(3)=0
9088 LET V(4)=60
9090 LET V(5)=0
9092 LET V(6)=200
9094 LET V(7)=100
9096 LET V(8)=0
9098 LET V(9)=100
9100 LET V(10)=120
9102 LET V(11)=0
9104 LET V(12)=140
9106 LET V(13)=150
9108 LET V(14)=140
9110 LET V(15)=160
9112 LET V(16)=200
9114 LET V(17)=180
9116 LET V(18)=0
9118 LET V(19)=180
9120 LET V(20)=200
9122 LET V(21)=0
9124 LET V(22)=220
9126 LET V(23)=0
9128 LET V(24)=220
9130 LET V(25)=240
9132 LET V(26)=200
9134 LET V(27)=260
9136 LET V(28)=260
9138 LET V(29)=150
9140 LET V(30)=280
9142 LET V(31)=0
9144 LET V(32)=300
9146 LET V(33)=300
9148 LET V(34)=0
9150 LET V(35)=320
9152 LET V(36)=200
9154 LET V(37)=0
9156 LET V(38)=350
9158 LET V(39)=0
9160 LET V(40)=400
9162 GOTO 100
```



```
PRESS ANY KEY TO CONTINUE
```



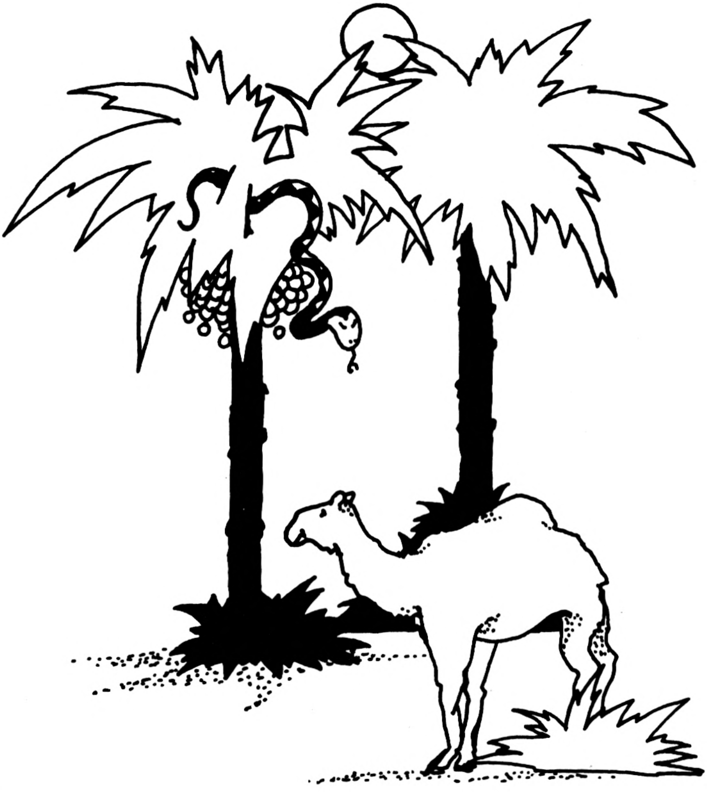
```
PRESS "NEWLINE" TO ROLL THE DICE
DICE =3
YOU LAND ON
WHITECHAPEL ROAD
WORTH £60
WILL YOU BUY WHITECHAPEL ROAD

FINE - THAT WILL BE £60
WILL YOU BUILD ON
WHITECHAPEL ROAD        ?
HOW MANY HOUSES?
AN HOTEL?
GOOD
YOU NOW HAVE £7940
I HAVE £9000
```



```
-----------------------------------
MY SHOT
DICE=6
I LANDED ON THE ANGEL ISLINGTON

I BUY        THE ANGEL ISLINGTON
 WITH 3 HOUSES
```

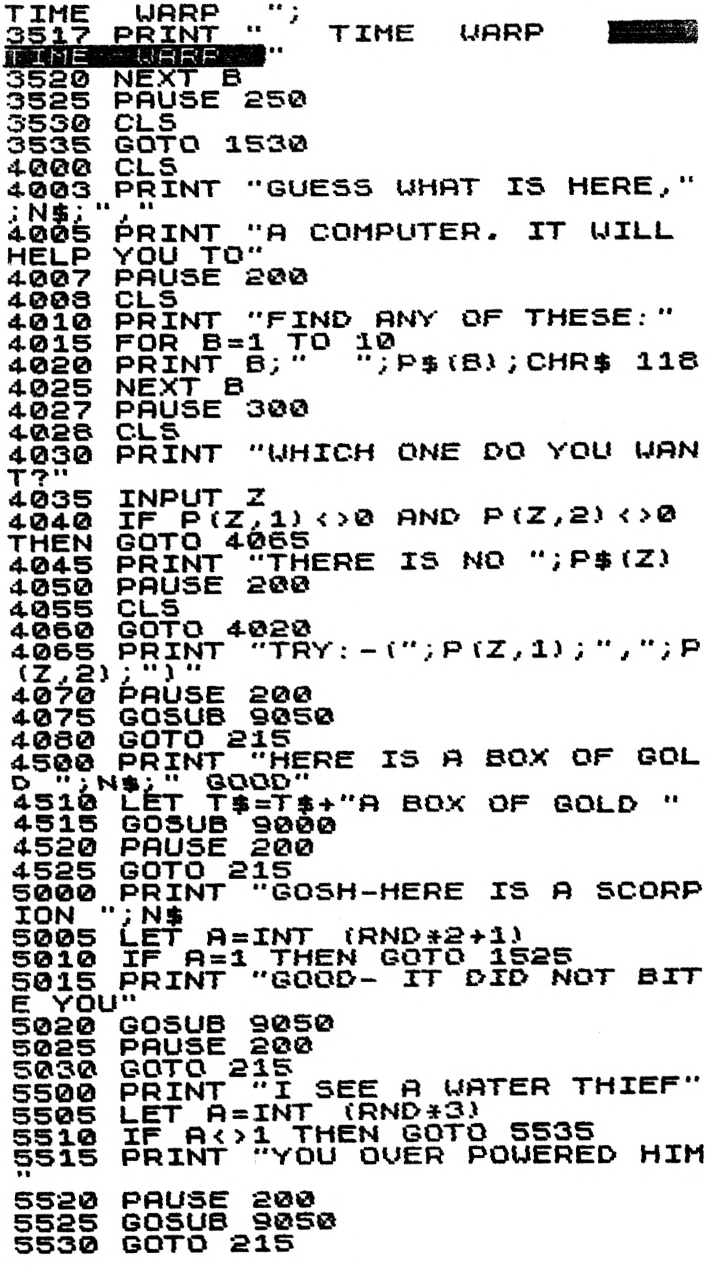
CAMEL

You are in the midst of the Sahara Desert, wandering about using the N, S, E and W keys (to go North, South, East and West respectively), finding treasure and avoiding poisonous snakes and water thieves, among a multitude of dangers. There are ten hidden hazards and treasures. Full instructions are included in the program, which was written by Chris Callender, and adapted by Clive Gifford.



```
   5 LET T$=""
  10 PRINT TAB 12;"▛▚▚▚▚▚▚▜"
  15 PRINT TAB 12;"▌CAMEL▐"
  20 PRINT TAB 12;"▙▚▚▚▚▚▚▟"
  25 PRINT
  30 PRINT
  35 PRINT "HELLO, WHAT IS YOUR
NAME?"
  40 INPUT N$
  45 PRINT
  50 PRINT "HELLO, ";N$;"."
  55 PRINT
  60 PRINT "DO YOU KNOW THE RULE
S?"
  65 INPUT R$
  70 IF R$(1)="N" THEN GOSUB 800
0
  75 LET W=200
  80 DIM R(200)
  85 FOR F=1 TO 10
  90 LET R(F)=F*500+500
  95 NEXT F
  96 FOR F=11 TO 200
  97 LET R(F)=R((F-10))
 100 NEXT F
 105 DIM P$(200,20)
 110 DIM P(200,2)
 115 DIM S$(20,32)
 120 FOR B=1 TO 100
 125 LET P(B,1)=INT (RND*27)+2
 130 LET P(B,2)=INT (RND*17)+2
 135 NEXT B
 140 LET P$(1)="A SWIRLING SAND
STORM"
 145 LET P$(2)="A POISONOUS SNAK
E"
 150 LET P$(3)="A LARGE OASIS"
 155 LET P$(4)="THE LONG LOST JE
WELS"
 160 LET P$(5)="A CAMEL"
 165 LET P$(6)="A TIME WARP"
 170 LET P$(7)="A COMPUTER"
 175 LET P$(8)="A BOX OF GOLD"
 180 LET P$(9)="A SCORPION"
 185 LET P$(10)="WATER THEIVES"
 190 FOR F=11 TO 200
 195 LET P$(F)=(P$(F-10))
 200 NEXT F
 205 LET X=INT (RND*27)+2
 210 LET Y=INT (RND*18)+2
 215 LET S$(1)="1   5    10    15
   20    25    30 "
 220 LET S$(20)="80
                  "
 225 FOR B=2 TO 19
 230 LET S$(B)="█
               █"
 232  NEXT B
 234  FOR F=5 TO 19 STEP 5
 236  LET S$(F,1)="▤"
 238  NEXT F
 240  LET B$="▓"
 245  FOR B=2 TO X-1
 250  LET B$=B$+" "
 255  NEXT B
 260  LET B$=B$+"*"
 265  FOR B=X+1 TO 31
 270  LET B$=B$+" "
 275  NEXT B
 280  LET B$=B$+"▓"
 285  LET S$(Y)=B$
 290  LET S$(18)="▓
               A  ▓"
 295  CLS
 300  FOR B=1 TO 20
 305  PRINT S$(B)
 310  NEXT B
 315  LET N=1
 320  IF P(N,1)=ABS X AND P(N,2)=
ABS Y THEN GOTO 420
 325  LET N=N+1
 330  IF N<>201 THEN GOTO 320
 335  IF ABS Y=18 AND ABS X=29 TH
EN GOTO 7000
 340  PRINT "DIRECTION?"
 345  INPUT D$
 350  LET C$=D$(2 TO (LEN D$))
 355  LET C=VAL (C$)
 360  IF C>10 THEN GOTO 345
 365  LET D$=D$(1)
 370  IF D$="N" THEN LET Y=Y-C
 375  IF D$="S" THEN LET Y=Y+C
 380  IF D$="E" THEN LET X=X+C
 385  IF D$="W" THEN LET X=X-C
 390  IF X=30 AND Y=19 THEN GOTO
7000
 395  IF W<10 THEN PRINT "GET WAT
ER FAST"
 400  PAUSE 100
 405  LET W=W-1
 410  IF W=0 THEN GOTO 6000
 415  GOTO 215
 420  GOTO R(N)
1000  CLS
1005  PRINT "YOU END UP IN A SAND
 STORM. YOU"
1010  PRINT "CANNOT SEE AND WANDE
R TO";
1015  LET X=INT (RND*28)+2
1020  LET Y=INT (RND*19)+2
1025  PRINT "(";X;";";Y;")"
1030  PAUSE 100
1035  GOSUB 9050
1040  GOTO 215
```

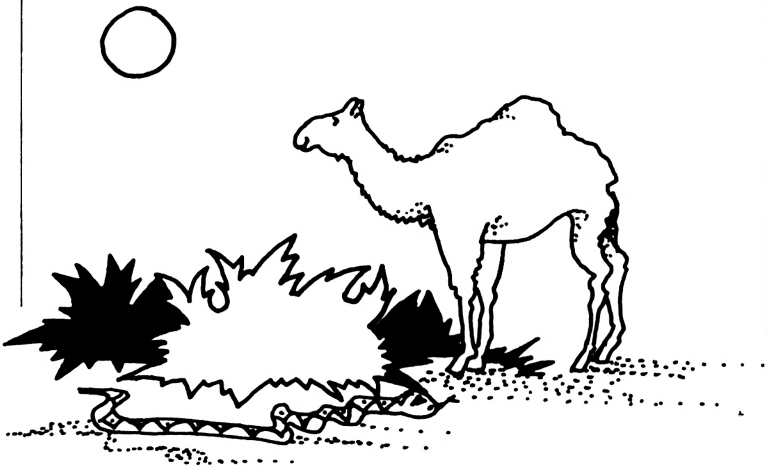
```
1500 CLS
1505 PRINT "OH DEAR; HERE IS A S
NAKE. I HATE"
1510 PRINT "SNAKES. HIT IT PLEAS
E"
1515 LET A=INT (RND*5)
1520 IF A=1 THEN GOTO 1550
1525 PRINT "OH NO - IT BIT YOU."
1530 PAUSE 200
1531 CLS
1532 PRINT "ANOTHER GAME?"
1535 INPUT G$
1540 IF G$(1)="Y" THEN RUN
1545 GOTO 9999
1550 PRINT "YOU WERE LUCKY THERE
 . BE CAREFUL"
1555 GOSUB 9050
1560 PAUSE 250
1565 GOTO 215
2000 PRINT "GREAT -I THINK THERE
 IS WATER."
2010 LET A=INT (RND*2)+1
2015 IF A=2 THEN GOTO 2060
2020 PRINT "YES IT IS. TAKE100 P
INTS"
2030 LET W=W+100
2040 GOTO 2065
2050 GOTO 9999
2060 PRINT "NO -IT WAS A MIRAGE.
 SORRY."
2065 PAUSE 250
2070 GOSUB 9050
2075 GOTO 215
2500 PRINT "GREAT-HERE ARE THE L
ONG LOST"
2505 PRINT "JEWELS"
2510 LET T$=T$+"THE LONG LOST JE
WELS"
2520 GOSUB 9000
2525 PAUSE 250
2530 GOTO 215
3000 PRINT "GOOD, ";N$;" HERE IS
 A CAMEL"
3004 PAUSE 250
3006 CLS
3010 PRINT "WHERE DO YOU WANT TO
 TAKE IT TO?"
3012 PRINT "ENTER ""X"" THEN NEW
 LINE THEN ""Y"""
3020 INPUT X
3025 INPUT Y
3030 GOSUB 9050
3035 GOTO 215
3500 PAUSE 250
3505 CLS
3510 FOR B=1 TO 10
3515 PRINT "    TIME   WARP    
```



```
TIME  WARP  ";
3517 PRINT "   TIME  WARP   
TIME  WARP  "
3520 NEXT B
3525 PAUSE 250
3530 CLS
3535 GOTO 1530
4000 CLS
4003 PRINT "GUESS WHAT IS HERE,"
;N$;","
4005 PRINT "A COMPUTER. IT WILL
HELP YOU TO"
4007 PAUSE 200
4008 CLS
4010 PRINT "FIND ANY OF THESE:"
4015 FOR B=1 TO 10
4020 PRINT B;"  ";P$(B);CHR$ 118
4025 NEXT B
4027 PAUSE 300
4028 CLS
4030 PRINT "WHICH ONE DO YOU WAN
T?"
4035 INPUT Z
4040 IF P(Z,1)<>0 AND P(Z,2)<>0
THEN GOTO 4065
4045 PRINT "THERE IS NO ";P$(Z)
4050 PAUSE 200
4055 CLS
4060 GOTO 4020
4065 PRINT "TRY:-(";P(Z,1);",";P
(Z,2);")"
4070 PAUSE 200
4075 GOSUB 9050
4080 GOTO 215
4500 PRINT "HERE IS A BOX OF GOL
D ";N$;" GOOD"
4510 LET T$=T$+"A BOX OF GOLD "
4515 GOSUB 9000
4520 PAUSE 200
4525 GOTO 215
5000 PRINT "GOSH-HERE IS A SCORP
ION ";N$
5005 LET A=INT (RND*2+1)
5010 IF A=1 THEN GOTO 1525
5015 PRINT "GOOD- IT DID NOT BIT
E YOU"
5020 GOSUB 9050
5025 PAUSE 200
5030 GOTO 215
5500 PRINT "I SEE A WATER THIEF"
5505 LET A=INT (RND*3)
5510 IF A<>1 THEN GOTO 5535
5515 PRINT "YOU OVER POWERED HIM
"
5520 PAUSE 200
5525 GOSUB 9050
5530 GOTO 215
```

```
5535 LET Q=INT (RND*W)
5540 PRINT "HE STOLE ";Q;" PINTS
"
5545 LET W=W-Q
5550 PAUSE 200
5552 GOSUB 9000
5560 GOTO 120
6000 PRINT "YOU DIED OF THIRST,"
;N$
6005 PRINT "DID YOU KNOW THAT???
"
6010 PAUSE 200
6015 CLS
6020 GOTO 1530
7000 PRINT "YOU GOT OUT OF MY DE
SERT ALIVE"
7010 IF T$="" THEN GOTO 7040
7020 PRINT "AND MANAGED TO TAKE:
-"
7022 PAUSE 200
7025 CLS
7030 PRINT T$
7040 PRINT "PREPARE TO TAKE OFF"
7045 GOTO 1530
8000 CLS
8002 PRINT
8003 PRINT
8010 PRINT "WELCOME TO THE GAME
CAMEL,"
8015 PRINT N$;"."
8020 PRINT "THE AIM OF CAMEL IS
TO GO ABOUT"
8030 PRINT "THE DESERT USING KEY
S N,S,E,+W"
8040 PRINT "FOR DIRECTION, AND A
 NUMBER LESS"
8050 PRINT "THAN 6 AFTER THE N,S
,E OR W"
8060 PRINT "TO TELL THE ZX81 THE
 NUMBER OF"
8070 PRINT "MILES YOU WANT TO MO
VE IN THAT"
8080 PRINT "DIRECTION. E5=5 MILE
S EAST. IN"
8090 PRINT "THE DESERT ARE 10 HI
DDEN HAZARDS"
8100 PRINT " AND TREASURES. YOU
MUST COLLECT"
8110 PRINT "AS MUCH TREASURE AS
YOU CAN AND "
8120 PRINT "GET TO THE AIRPORT.W
HICH IS THE"
8130 PRINT " A IN THE BOTTOM RIG
HT HAND"
8140 PRINT "CORNER."
8145 PRINT "BUT BEWARE OF LINE 1
8...THERE IS"
```

```
8147 PRINT "A STRONG HEATHAZE WH
ICH MAKES  "
8148 PRINT "YOUR POSITION IMPOSS
IBLE TO SEE."
8149 PRINT
8150 PRINT "     ***HAVE A GOOD
GAME***"
8160 PAUSE 1500
8180 RETURN
9000 IF N<=10 THEN GOTO 9015
9005 LET N=N-10
9010 GOTO 9000
9015 FOR F=N TO 200 STEP 10
9020 LET P(F,1)=0
9025 LET P(F,2)=0
9030 NEXT F
9035 RETURN
9050 LET P(N,1)=0
9055 LET P(N,2)=0
9060 RETURN
9999 STOP
```

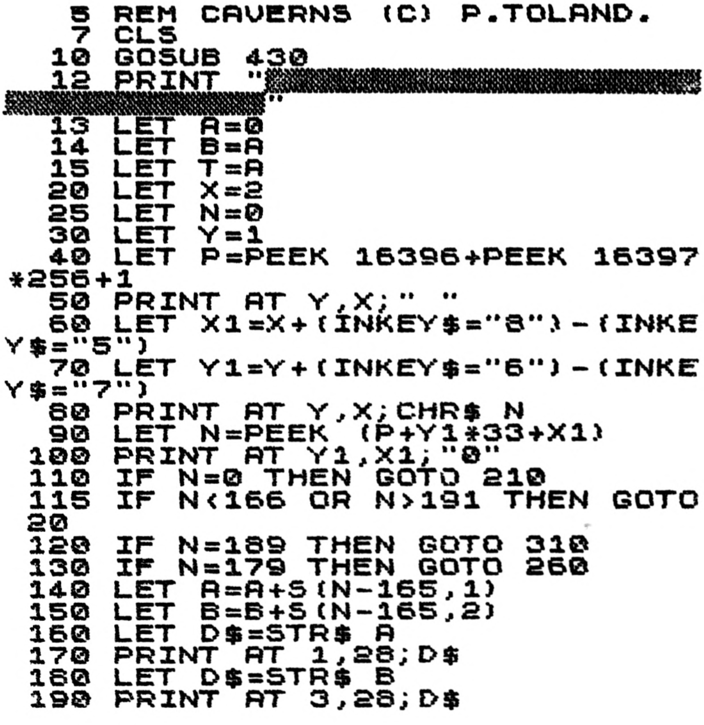


MIND BENDERS

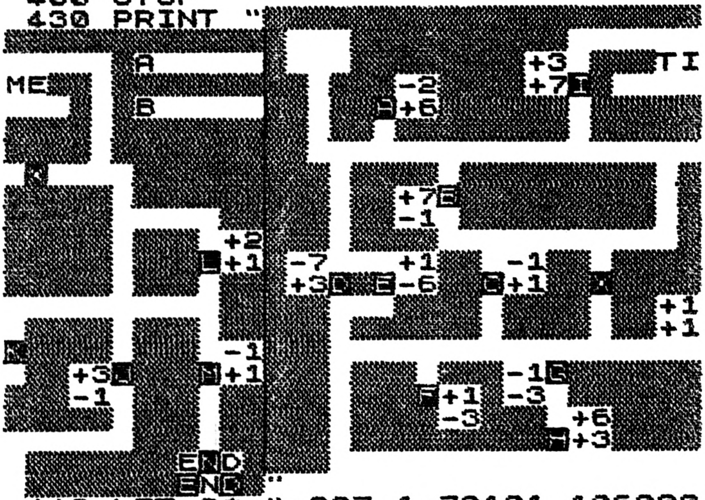
CAVERNS

Paul Toland's marvellous graphics game Caverns need not destroy your nerves, so long as you concentrate. This game is a test of mental arithmetic applied to a specific problem. You must get to the end of the network of curves using the path of your choice. The catch is that at several points in the caves you have your two totals, A and B, altered by the displayed amounts.

You must reset A and B to zero again before you can safely enter the end of the caves. Hitting the walls puts you back at the start, as does hitting an inverse X. At the end of the game you will be offered another round. Just press ENTER if you want another game, 'N' if you do not.



```
   5 REM CAVERNS (C) P.TOLAND.
   7 CLS
  10 GOSUB 430
  12 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒"
  13 LET A=0
  14 LET B=A
  15 LET T=A
  20 LET X=2
  25 LET N=0
  30 LET Y=1
  40 LET P=PEEK 16396+PEEK 16397
*256+1
  50 PRINT AT Y,X;" "
  60 LET X1=X+(INKEY$="8")-(INKE
Y$="5")
  70 LET Y1=Y+(INKEY$="6")-(INKE
Y$="7")
  80 PRINT AT Y,X;CHR$ N
  90 LET N=PEEK (P+Y1*33+X1)
 100 PRINT AT Y1,X1;"0"
 110 IF N=0 THEN GOTO 210
 115 IF N<166 OR N>191 THEN GOTO
 20
 120 IF N=189 THEN GOTO 310
 130 IF N=179 THEN GOTO 260
 140 LET A=A+S(N-165,1)
 150 LET B=B+S(N-165,2)
 160 LET D$=STR$ A
 170 PRINT AT 1,28;D$
 180 LET D$=STR$ B
 190 PRINT AT 3,28;D$
 210 LET X=X1
 220 LET Y=Y1
 230 LET T=T+1
 240 PRINT AT 3,16;T
 250 GOTO 50
 260 IF A<>0 OR B<>0 THEN GOTO 2
90
 270 PRINT "YOU MADE IT - WHAT K
EPT YOU ?"
 280 GOTO 350
 290 PRINT "HARD LUCK - A AND B
MUST BE ZERO"
 300 GOTO 350
 310 PRINT "YOU HIT AN ▨"
 350 PRINT ,"TRY AGAIN?"
 360 INPUT C$
 370 IF C$<>"N" THEN RUN
 400 STOP
 430 PRINT "
```

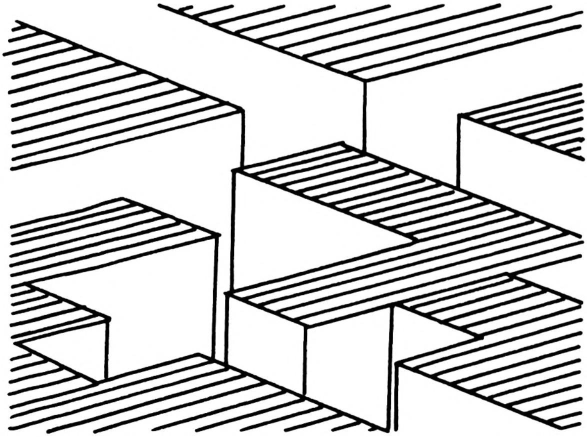


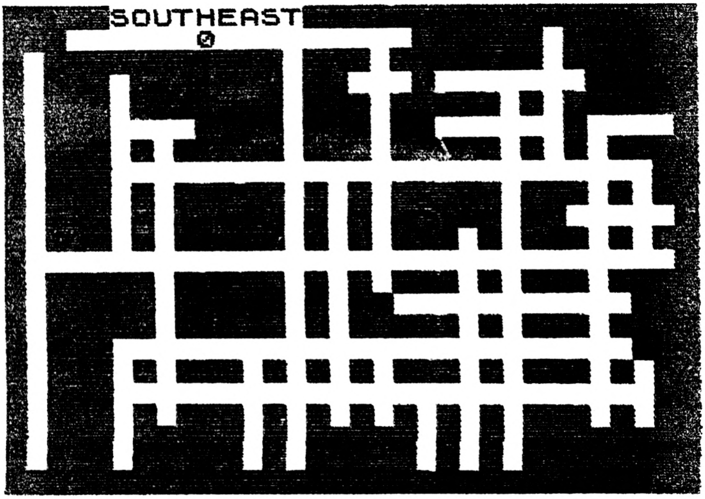
```
 440 LET A$="-207-1-70101-106030
30102-1"
 450 LET B$="06-10103-6-3-30307-
1010101"
 460 DIM S(13,2)
 470 FOR I=1 TO 13
 480 LET S(I,1)=VAL A$(I*2-1 TO
I*2)
 490 LET S(I,2)=VAL B$(I*2-1 TO
I*2)
 500 NEXT I
 510 DIM D$(4)
 520 RETURN
```

HURKLE'S MAZE

In this game, written by Paul Toland, you are looking for the Hurkle on a randomly generated maze-type board, searching in real time. You move through the maze using the '5', '6', '7' and '8' keys, and your task is to try and land on the square occupied by the Hurkle. You cannot, however, see the Hurkle (they are invisible except under dilithium light), so you get clues as to the creature's current position in terms of North, South, East and West. When you have found the Hurkle, press 'N' to stop, or any other key for a new game.

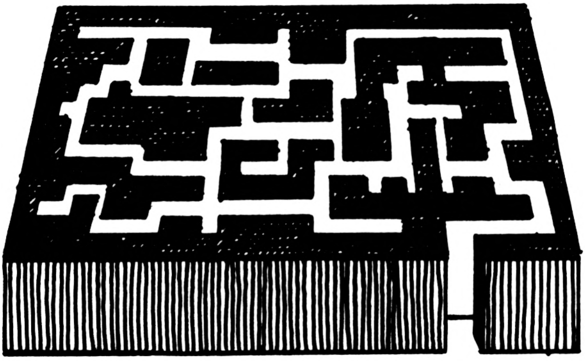
Line 5 modifies the random maze generation to ensure that all parts of the maze can be accessed. The characters in M$ have codes which generate usable screens when put into a RAND N instruction. There are many such screens, but we have only given 14 of the best of them. The maze generation process runs in the SLOW mode and, although it takes a while, is fascinating to watch.





```
SOUTHEAST
    0
```

```
   2 RAND
   5 LET M$="P▝▖?>/49EHIKLN"
   7 RAND CODE M$(INT (RND*14)+1
)
  10 FOR I=1 TO 22
  20 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒"
  30 NEXT I
  40 FOR I=1 TO 30
  50 LET X=INT (RND*28)+1
  60 LET Y=INT (RND*9)*2+1
  70 LET L=INT (RND*(29-X))+2
  80 FOR J=0 TO L
  90 PRINT AT Y,X+J;" "
 100 NEXT J
 110 LET X=INT (RND*15)*2+1
 120 LET Y=INT (RND*17)+1
 130 LET L=INT (RND*(19-Y))+2
 140 FOR J=0 TO L
 150 PRINT AT Y+J,X;" "
 160 NEXT J
 170 NEXT I
 175 LET T=0
 180 LET P=PEEK 16396+256*PEEK 1
6397+1
 185 RAND
 190 LET X=INT (RND*704)+1
 195 IF PEEK (P+X)>0 THEN GOTO 1
90
 200 LET H=INT (RND*704)+1
 205 IF PEEK (P+H)>0 THEN GOTO 2
00
 210 POKE P+X,28
 220 LET D=X+(INKEY$="8")-(INKEY
$="5")+(INKEY$="6")*33-(INKEY$="
7")*33
 230 POKE P+X,0
 240 LET T=T+1
 250 IF PEEK (P+D)=0 THEN LET X=
D
 255 IF X=H THEN GOTO 350
 260 LET XD=INT (X/33)
 265 LET XA=X-XD*33
 270 LET HD=INT (H/33)
 275 LET HA=H-HD*33
 277 PRINT AT 0,5;"█████████████
█"
 280 IF XD=HD THEN GOTO 310
 290 IF HD<XD THEN PRINT AT 0,5;
"NORTH."
 300 IF HD>XD THEN PRINT AT 0,5;
"SOUTH."
 310 IF XA=HA THEN GOTO 210
 320 IF HA<XA THEN PRINT AT 0,10
;"WEST"
 330 IF HA>XA THEN PRINT AT 0,10
;"EAST"
 340 GOTO 210
 350 PRINT "YOU FOUND HURKLE IN
";T;" SECS.",,"ANOTHER GO?"
 360 POKE P+H,45
 370 IF INKEY$="" THEN GOTO 370
 380 IF INKEY$="N" THEN GOTO 410
 390 CLS
 400 RUN
 410 PRINT ,"GOODBYE"
 420 GOTO 410
```



EXECUTION

In this snappy variation of Hangman, by Paul Toland, you store a vocabulary of 30 words within the program, and the computer chooses at random from one of the words. You then have a limited time to guess the computer's words.

When you first RUN it, you will be asked (line 3) to enter your words, one by one. You may prefer to have someone else do this, so you do not know which words are being entered. You will find the game is more fun if you enter words which bear some similarity to each other (like FRIED, FIEND, FRIEND and TRIED). Once you have entered the words they can be saved on tape along with the program.

You save the program by entering GOTO 390. This will ensure that, on reloading, the program will start automatically, thus protecting your words from being wiped out with the use of RUN. If you want to RUN it after the program has been halted, start it with GOTO 5.

```
   1 DIM D$(30,10)
   2 FOR I=1 TO 30
   3 INPUT D$(I)
   4 NEXT I
   5 PRINT AT 2,10;"EXECUTION"
   6 REM (C) PAUL TOLAND
   7 PRINT AT 0,0;
  10 RAND
  20 LET G=0
  30 LET C=G
  40 FOR I=1 TO 10
  50 PRINT I;TAB 2;"█   █"
  60 NEXT I
  70 PRINT "  ▒▒▒▒▒"
  80 PRINT AT 8,3;"O O"
  90 PRINT AT 9,3;">=<"
  95 PRINT AT 19,3;"ENTER GUESS"
 100 LET R=INT (RND*30)+1
 110 LET W$=""
 120 FOR I=1 TO 10
 130 IF D$(R,I)>" " THEN LET W$=
W$+D$(R,I)
 140 NEXT I
```

```
150 LET L=LEN W$
160 PRINT AT 8,20;"THE WORD"
165 LET C$="??????????"( TO L)
170 PRINT AT 10,20;C$
180 PRINT AT G,3;"▒▒▒"
190 IF INKEY$="" THEN GOTO 190
195 LET G$=INKEY$
200 FOR I=1 TO L
210 IF G$<>W$(I) THEN GOTO 240
220 LET G=G-1*(G>=0)
230 LET C$(I)=W$(I)
240 NEXT I
250 IF C$=W$ THEN GOTO 320
260 PRINT AT G,3;"   "
270 LET G=G+1
280 IF G<10 THEN GOTO 170
290 PRINT AT 12,3;"+ +"
300 PRINT "    -"
310 PRINT AT 2,20;"COMPLETED"
315 GOTO 330
320 PRINT AT 2,20;"SUSPENDED"
330 PRINT AT 10,20;W$
340 PRINT ,"TRY AGAIN?"
350 INPUT A$
360 IF A$="N" THEN STOP
370 CLS
380 GOTO 5
390 SAVE "EXEC"
400 GOTO 5
```

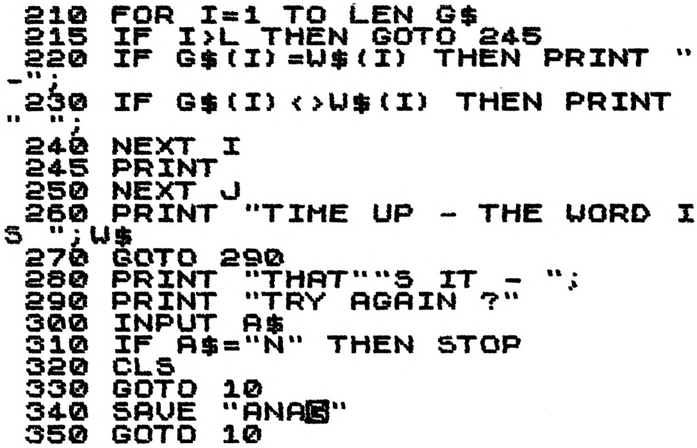
ANAGRAMS

Like Execution, this word game from Paul Toland allows you to store a vocabulary of 30 words within the program, with the computer choosing at random from the words. You have a limited number of guesses to work out the word.

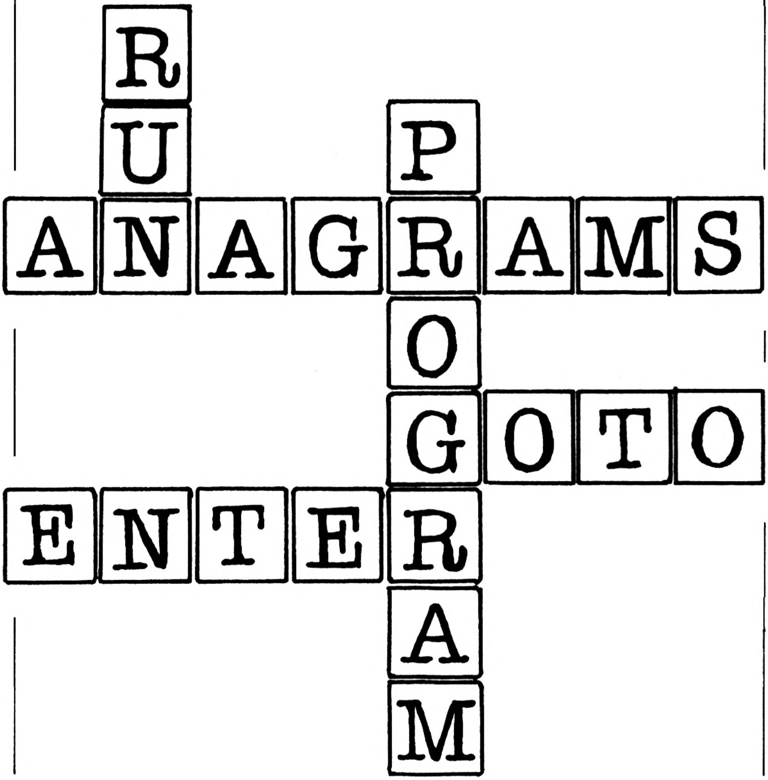
When you run the program, you will be asked to enter 30 words, pressing ENTER after each one. Then the program is ready. Run it, and you'll see the jumbled word; then you have to enter what you think the right word is. The correct letters will be highlighted for you.

Once you have the vocabulary in place, save the program with the command GOTO 340. This will ensure it will run automatically on loading. If you stop the program after loading it, and want to run it without losing your words, start it with GOTO 10.

```
   1 DIM D$(30,10)
   2 FOR I=1 TO 30
   3 INPUT D$(I)
   4 NEXT I
   5 REM (C) PAUL TOLAND
  10 RAND
  20 PRINT "     ANAGRAMS"
  30 LET R=INT (RND*30)+1
  40 LET W$=""
  50 FOR I=1 TO 10
  60 IF D$(R,I)>"  " THEN LET W$=
W$+D$(R,I)
  70 NEXT I
  80 LET L=LEN W$
  90 LET S$="          "( TO L)
 100 FOR I=1 TO L
 110 LET R=INT (RND*L)+1
 120 IF S$(R)>"  " THEN GOTO 110
 130 LET S$(R)=W$(I)
 140 NEXT I
 150 PRINT "YOUR ANAGRAM IS ";S$
 160 FOR J=1 TO 9
 170 PRINT J;"   ";
 180 INPUT G$
 190 PRINT G$
 200 IF G$=W$ THEN GOTO 280
 205 PRINT "   ";
```



```
 210 FOR I=1 TO LEN G$
 215 IF I>L THEN GOTO 245
 220 IF G$(I)=W$(I) THEN PRINT "
-";
 230 IF G$(I)<>W$(I) THEN PRINT
" ";
 240 NEXT I
 245 PRINT
 250 NEXT J
 260 PRINT "TIME UP - THE WORD I
S ";W$
 270 GOTO 290
 280 PRINT "THAT""S IT - ";
 290 PRINT "TRY AGAIN ?"
 300 INPUT A$
 310 IF A$="N" THEN STOP
 320 CLS
 330 GOTO 10
 340 SAVE "ANAS"
 350 GOTO 10
```



SELF DISCOVERY

REACTION

This reaction timer, written in BASIC and machine code, comes from Alistair Miller. The program tests your reactions at various skill levels. When it is first run it is in demonstration mode and shows the various skill levels. The machine code occupies 77 bytes and is stored in the REM at line 1.

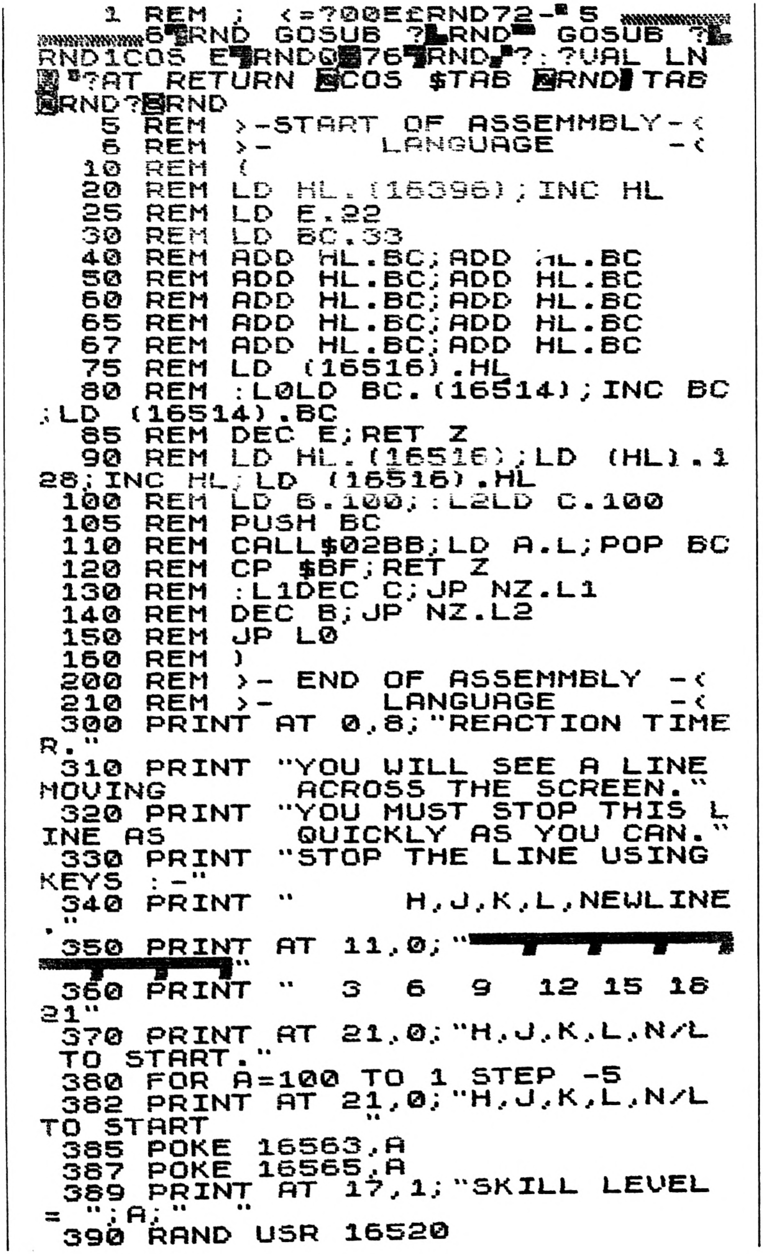
Once the program is working you enter your skill level. Then, after a random delay, a line starts to move across the screen at the specified speed. You now press ENTER/L/K/L or H to stop the line. You will be told how many blocks the line moved, and be given a chance to play the game again. If you press a key before the line has started — during the random delay — you'll be told you cheated, and the game will start again.

If you do not have an assembler you can omit lines 5 to 210. Assuming that you do not have an assembler, type in the word REM followed by 73 full stops. Then enter the routine from line 9000 onwards. Run the routine and enter the following, counting / as ENTER (note there are no spaces, but they are used here to make it easier to follow, and note also that these are zeros, not letter Os):

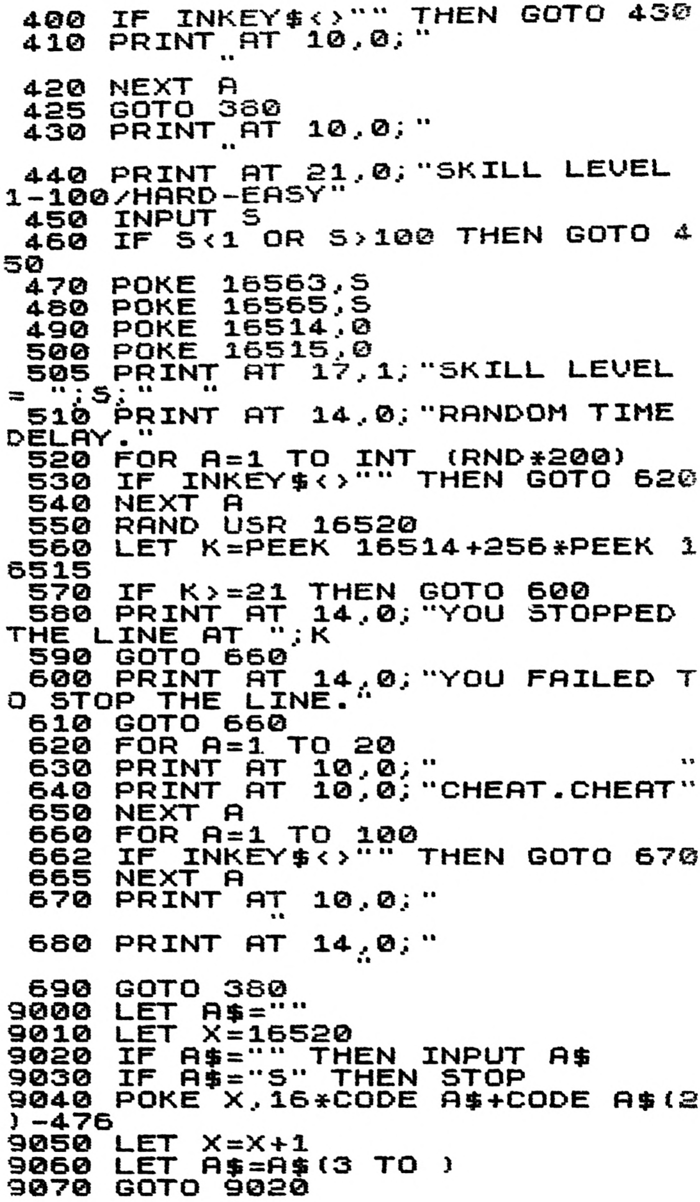
```
2A 0C 40 / 23 / IE 16 / 01 21 00 / 09 09 /
09 09 / 09 09 / 09 09 / 09 09 /
22 84 40 / ED 4B 82 40 / 03 /
ED 43 82 40 / ID / C8 / 2A 84 40 /
36 80 / 23 / 22 84 40 / 06 64 /
0E 64 / C5 / CD BB 02 / 7D / CI /
FE BF / C8 / 0D / C2 BF 40 / 05 /
C2 B4 40 / C3 9E 40 / S /
```

Now type in the rest of the program, then SAVE it before you run it.

REACTION



```
   1 REM : <=?00E£RND72-▘5 ▒▒▒▒▒▒▒
▒▒▒▒▒▒▒6▜RND GOSUB ?▙RND▀ GOSUB ?▙
RND1COS E▜RNDQ█76▜RND▞?:?VAL LN
▩▝?AT RETURN ▩COS $TAB ▩RND▌TAB
▩RND?▩RND
   5 REM >-START OF ASSEMMBLY-<
   6 REM >-      LANGUAGE      -<
  10 REM (
  20 REM LD HL.(16396);INC HL
  25 REM LD E.22
  30 REM LD BC.33
  40 REM ADD HL.BC;ADD HL.BC
  50 REM ADD HL.BC;ADD HL.BC
  60 REM ADD HL.BC;ADD HL.BC
  65 REM ADD HL.BC;ADD HL.BC
  67 REM ADD HL.BC;ADD HL.BC
  75 REM LD (16516).HL
  80 REM :L0LD BC.(16514);INC BC
;LD (16514).BC
  85 REM DEC E;RET Z
  90 REM LD HL.(16516);LD (HL).1
28;INC HL;LD (16516).HL
 100 REM LD B.100;:L2LD C.100
 105 REM PUSH BC
 110 REM CALL$02BB;LD A.L;POP BC
 120 REM CP $BF;RET Z
 130 REM :L1DEC C;JP NZ.L1
 140 REM DEC B;JP NZ.L2
 150 REM JP L0
 160 REM )
 200 REM >- END OF ASSEMMBLY -<
 210 REM >-      LANGUAGE      -<
 300 PRINT AT 0.8;"REACTION TIME
R."
 310 PRINT "YOU WILL SEE A LINE
MOVING        ACROSS THE SCREEN."
 320 PRINT "YOU MUST STOP THIS L
INE AS        QUICKLY AS YOU CAN."
 330 PRINT "STOP THE LINE USING
KEYS :-"
 340 PRINT "         H,J,K,L,NEWLINE
"
 350 PRINT AT 11,0;"▀▀▀▀▛▀▀▛▀▀▛▀▀▌
▀▀▛▀▀▛▀▀▌"
 360 PRINT "   3  6  9  12 15 18
21"
 370 PRINT AT 21,0;"H,J,K,L,N/L
 TO START."
 380 FOR A=100 TO 1 STEP -5
 382 PRINT AT 21,0;"H,J,K,L,N/L
TO START          "
 385 POKE 16563,A
 387 POKE 16565,A
 389 PRINT AT 17,1;"SKILL LEVEL
= ";A;"  "
 390 RAND USR 16520
```



```
 400 IF INKEY$<>"" THEN GOTO 430
 410 PRINT AT 10,0;"
          "
 420 NEXT A
 425 GOTO 380
 430 PRINT AT 10,0;"
          "
 440 PRINT AT 21,0;"SKILL LEVEL
1-100/HARD-EASY"
 450 INPUT S
 460 IF S<1 OR S>100 THEN GOTO 4
50
 470 POKE 16563,S
 480 POKE 16565,S
 490 POKE 16514,0
 500 POKE 16515,0
 505 PRINT AT 17,1;"SKILL LEVEL
= ";S;"  "
 510 PRINT AT 14,0;"RANDOM TIME
DELAY."
 520 FOR A=1 TO INT (RND*200)
 530 IF INKEY$<>"" THEN GOTO 620
 540 NEXT A
 550 RAND USR 16520
 560 LET K=PEEK 16514+256*PEEK 1
6515
 570 IF K>=21 THEN GOTO 600
 580 PRINT AT 14,0;"YOU STOPPED
THE LINE AT ";K
 590 GOTO 660
 600 PRINT AT 14,0;"YOU FAILED T
O STOP THE LINE."
 610 GOTO 660
 620 FOR A=1 TO 20
 630 PRINT AT 10,0;"            "
 640 PRINT AT 10,0;"CHEAT.CHEAT"
 650 NEXT A
 660 FOR A=1 TO 100
 662 IF INKEY$<>"" THEN GOTO 670
 665 NEXT A
 670 PRINT AT 10,0;"
            "
 680 PRINT AT 14,0;"
                "
 690 GOTO 380
9000 LET A$=""
9010 LET X=16520
9020 IF A$="" THEN INPUT A$
9030 IF A$="S" THEN STOP
9040 POKE X,16*CODE A$+CODE A$(2
)-476
9050 LET X=X+1
9060 LET A$=A$(3 TO )
9070 GOTO 9020
```

LIFE CHANGE UNITS

This program measures recent changes in your life and the effect those changes have on you. Any change in your life, be it good or bad, creates measurable stress. Christmas is stressful, and so is gaining a new job, or a new member of the family.

A number of situations — PREGNANCY for instance — will be printed on the screen. If you are pregnant then press the 'Y' (for YES) key. If not, press 'N'. At the end, an assessment of your situation will be made, and a comment made on this.

```
  10 REM L.C.U. (C)PAUL TOLAND
 150 PRINT "    L.C.U. TEST"
 160 PRINT " HOW CHANGES IN YOUR
 LIFESTYLE  AFFECT YOUR FUTURE"
 170 PRINT AT 12,6;"PRESS Y OR N
"
 180 LET L=0
 190 GOTO 250
 200 PRINT AT 10,0;L$;"
                              "
 210 IF INKEY$<>"Y" AND INKEY$<>
"N" THEN GOTO 210
 220 IF INKEY$="Y" THEN LET L=L+
VAL S$(I*2-1 TO I*2)
 230 LET I=I+1
 232 FOR K=1 TO 10
 234 NEXT K
 240 RETURN
 250 LET I=1
 260 LET S$="9973716563635350474
54544404039393938373635333130292
92929282626252423212020201919181
7161515131211"
 270 LET L$="DEATH OF SPOUSE"
 275 GOSUB 200
 280 LET L$="DIVORCE"
 285 GOSUB 200
 290 LET L$="GIVING UP HARD DRUG
S"
 295 GOSUB 200
 300 LET L$="MARITAL SEPERATION"
 305 GOSUB 200
 310 LET L$="PRISON SENTENCE"
 315 GOSUB 200
 320 LET L$="DEATH OF CLOSE FAMI
LY MEMBER"
 325 GOSUB 200
 330 LET L$="PERSONAL INJURY OR
ILLNESS"
 335 GOSUB 200
 340 LET L$="MARRIAGE"
 345 GOSUB 200
 350 LET L$="SACKED / MADE REDUN
DANT"
 355 GOSUB 200
 360 LET L$="MARITAL RECONCILIAT
ION"
 365 GOSUB 200
 370 LET L$="RETIREMENT"
 375 GOSUB 200
 380 LET L$="CHANGE IN A FAMILY
MEMBERS HEALTH"
 385 GOSUB 200
 390 LET L$="PREGNANCY"
 395 GOSUB 200
 400 LET L$="GIVING UP > 40 CIGA
RETTES A DAY"
 405 GOSUB 200
 410 LET L$="SEX DIFFICULTIES"
 415 GOSUB 200
 420 LET L$="GAIN OF FAMILY MEMB
ER"
 425 GOSUB 200
 430 LET L$="BUSINESS READJUSTME
NT"
 435 GOSUB 200
 440 LET L$="CHANGE IN FINANCIAL
 STATE"
 445 GOSUB 200
 450 LET L$="DEATH OF CLOSE FRIE
ND"
 455 GOSUB 200
 460 LET L$="CHANGE TO DIFFERENT
 LINE OF WORK"
 465 GOSUB 200
 470 LET L$="CHANGE IN NO. OF AR
GUMENTS WITH SPOUSE"
 475 GOSUB 200
 480 LET L$="PRE-MENSTRUAL TENSI
ON"
 485 GOSUB 200
 490 LET L$="FORECLOSURE OF MORT
GAGE OR LOAN"
 495 GOSUB 200
 500 LET L$="MORTGAGE OVER £20,0
00"
 505 GOSUB 200
 510 LET L$="CHANGE IN WORK RESP
ONSIBILITY"
 515 GOSUB 200
 520 LET L$="JET LAG"
 525 GOSUB 200
 530 LET L$="SON OR DAUGHTER LEA
VING HOME"
 535 GOSUB 200
 540 LET L$="TROUBLE WITH IN-LAW
S"
 545 GOSUB 200
 550 LET L$="OUTSTANDING PERSONA
L ACHIEVEMENT"
 555 GOSUB 200
 560 LET L$="WIFE BEGINS OR STOP
S WORK"
 565 GOSUB 200
 570 LET L$="CHILDREN START OR S
TOP SCHOOL"
 575 GOSUB 200
 580 LET L$="CHANGE IN LIVING CO
NDITIONS"
 585 GOSUB 200
 590 LET L$="REVISION OF PERSONA
L HABITS"
 595 GOSUB 200
 600 LET L$="TROUBLE WITH BOSS"
 605 GOSUB 200
 610 LET L$="GIVING UP < 40 CIGA
RETTES A DAY"
 615 GOSUB 200
 620 LET L$="CHANCE IN WORK HOUR
S OR CONDITIONS"
 625 GOSUB 200
 630 LET L$="CHANGE IN RESIDENCE
"

 635 GOSUB 200
 640 LET L$="CHANGE IN SCHOOL"
 645 GOSUB 200
 650 LET L$="CHANGE IN RECREATIO
N"
 655 GOSUB 200
 660 LET L$="CHANGE IN CHURCH AC
TIVITY"
 665 GOSUB 200
 670 LET L$="CHANGE IN SOCIAL AC
TIVITY"
 675 GOSUB 200
 680 LET L$="MORTGAGE OR LOAN UN
DER £20,000"
 682 GOSUB 200
 684 LET L$="CHANGE IN SLEEPING
HABITS"
 686 GOSUB 200
 690 LET L$="CHANGE IN NO. OF FA
MILY GET-TOGETHERS"
```

```
 695 GOSUB 200
 700 LET L$="CHANGE IN EATING HA
BITS"
 705 GOSUB 200
 710 LET L$="GOING ON A HOLIDAY"
 715 GOSUB 200
 720 LET L$="IT IS CHRISTMAS"
 725 GOSUB 200
 730 LET L$="MINOR VIOLATION OF
THE LAW"
 735 GOSUB 200
 740 PRINT AT 7,3;"YOUR L.C.U. T
OTAL IS ";L
 750 PRINT ,,"       THE VERDICT-"
 760 IF L<150 THEN PRINT "YOU AR
E LIVING A STABLE , SAFE  WAY OF
 LIFE AND YOU ARE LESS    LIKELY
 THAN AVERAGE TO HAVE AN   ACCID
ENT OR BECOME ILL."
 765 IF L=0 THEN PRINT "---UNLES
S YOU DIE OF BOREDOM"
 770 IF L>149 AND L<200 THEN PRI
NT "THERE IS A 37 PER CENT CHANC
E    OF YOU ENCOUNTERING HEALTH O
R    SAFEY PROBLEMS."
 780 IF L>199 AND L<300 THEN PRI
NT "YOU ARE 51 PER CENT LIKELY T
O    HAVE HEALTH OR SAFETY PROBL
EMS;THIS IS A GOOD TIME TO CALM
DOWN"
 790 IF L>299 THEN PRINT "**WATC
H OUT** YOU NOW HAVE A 79 PER CE
NT CHANCE OF HAVING HEALTH OR SA
FETY PROBLEMS. DO NOT GIVEUP HOP
E-SIMPLY TAKE EXTRA CARE  AND LE
AD A QUIET LIFE UNTIL YOURSCORE
DROPS."
 800 PRINT ,,"ANY MORE ""CLIENTS
""?"
 810 INPUT A$
 820 IF A$="N" THEN STOP
 830 CLS
 840 GOTO 1
```

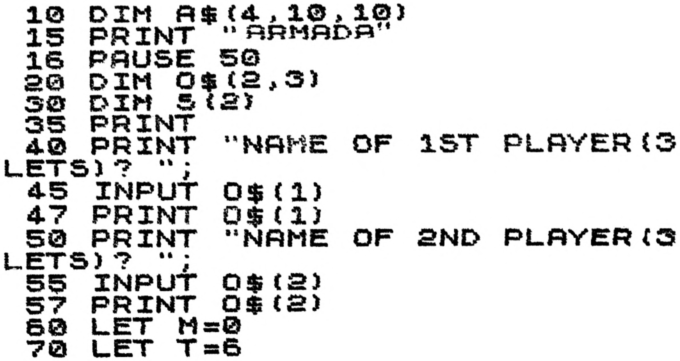
THE ZX81 TAKES CONTROL

ARMADA

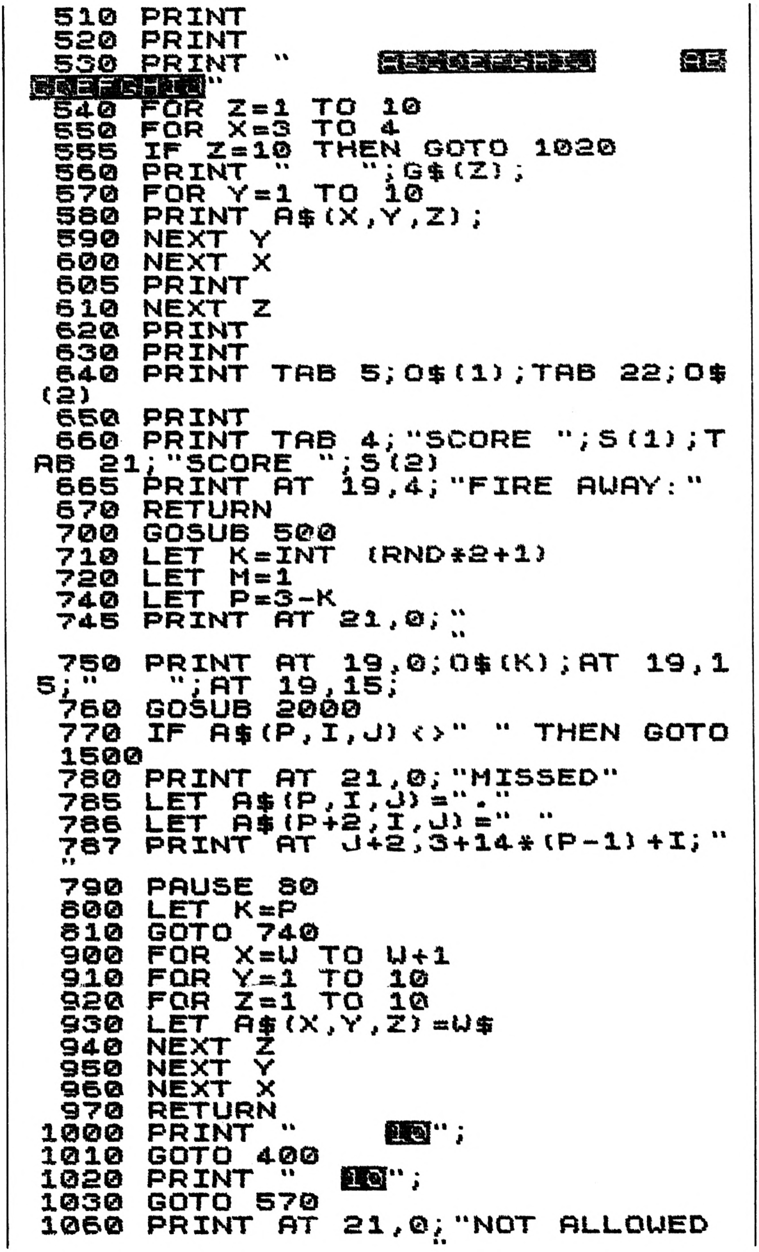
This is an elaborate seafaring war game for two players, written by Jim Archer. Although it is based on Battleships, it has many developments not found in the standard game. For a start there are six different types of ships: six battleships, five cruisers, four torpedo boats, three aircraft carriers, two submarines and one destroyer. The fewer there are of them, the more points you gain by hitting one.

The players are first asked to give their fleet positions (while the other player averts his eyes) to the computer, which acts as umpire over the forthcoming sea battle. Positions entered are checked; they *must* be adjacent to another ship of their own kind, and *not* next to a ship of a different kind.

A player is then chosen by the computer to start, and the players take turns at firing onto the opponent's territory. A hit gives you an extra turn, and as each square is tried it is 'whited out' so you will know not to try that square again. If, however, you do try a square again the computer will let you know with the words TRIED ALREADY. If you crash the program (which is pretty hard to do since the computer will query odd inputs) you can restore the current game with GOTO 2500.



```
  10 DIM A$(4,10,10)
  15 PRINT "ARMADA"
  16 PAUSE 50
  20 DIM O$(2,3)
  30 DIM S(2)
  35 PRINT
  40 PRINT "NAME OF 1ST PLAYER(3
LETS)? ";
  45 INPUT O$(1)
  47 PRINT O$(1)
  50 PRINT "NAME OF 2ND PLAYER(3
LETS)? ";
  55 INPUT O$(2)
  57 PRINT O$(2)
  60 LET M=0
  70 LET T=6
  80 LET W=1
  90 LET W$="  "
  95 LET G$="123456789"
  97 FAST
 100 GOSUB 900
 105 SLOW
 110 LET H=0
 120 GOSUB 1100
 130 FOR X=1 TO 2
 140 LET S(X)=0
 155 GOSUB 350
 160 FOR R=T TO 1 STEP -1
 200 FOR U=1 TO R
 205 LET H=H+7-R
 207 PRINT AT 21,0;R;"  ";B$(R,4
TO 3+VAL (B$(R,2 TO 3)));
 208 IF R>1 THEN PRINT "S";
 209 PRINT " :          ";
 210 PRINT AT 21,21;"NO.";U;"? "
;AT 21,28;"    ";AT 21,28;
 220 GOSUB 2000
 225 GOTO 1700
 230 LET A$(X,I,J)=B$(R,4)
 235 PRINT AT 2*J,5+2*I;B$(R,4)
 240 NEXT U
 270 NEXT R
 280 NEXT X
 300 LET W=3
 310 LET W$="▒"
 315 FAST
 320 GOSUB 900
 325 SLOW
 330 GOTO 700
 350 CLS
 351 PRINT O$(X);" GIVE YOUR FLE
ET POSITIONS"
 352 PAUSE 100
 353 CLS
 355 FAST
 360 PRINT O$(X);"     A B C D E
F G H I J"
 365 PRINT
 370 FOR Z=1 TO 10
 380 IF Z=10 THEN GOTO 1000
 390 PRINT "        ";G$(Z);
 400 FOR Y=1 TO 10
 410 IF A$(X,Y,Z)<>" " THEN PRIN
T TAB (2*Y+1);A$(X,Y,Z);
 420 NEXT Y
 398 PRINT
 440 PRINT
 450 NEXT Z
 470 SLOW
 480 RETURN
 500 CLS
```



```
 510 PRINT
 520 PRINT
 530 PRINT "        ABCDEFGHIJ     AB
CDEFGHIJ"
 540 FOR Z=1 TO 10
 550 FOR X=3 TO 4
 555 IF Z=10 THEN GOTO 1020
 560 PRINT "    ";G$(Z);
 570 FOR Y=1 TO 10
 580 PRINT A$(X,Y,Z);
 590 NEXT Y
 600 NEXT X
 605 PRINT
 610 NEXT Z
 620 PRINT
 630 PRINT
 640 PRINT TAB 5;O$(1);TAB 22;O$
(2)
 650 PRINT
 660 PRINT TAB 4;"SCORE ";S(1);T
AB 21;"SCORE ";S(2)
 665 PRINT AT 19,4;"FIRE AWAY:"
 670 RETURN
 700 GOSUB 500
 710 LET K=INT (RND*2+1)
 720 LET M=1
 740 LET P=3-K
 745 PRINT AT 21,0;"
"
 750 PRINT AT 19,0;O$(K);AT 19,1
5;"    ";AT 19,15;
 760 GOSUB 2000
 770 IF A$(P,I,J)<>" " THEN GOTO
 1500
 780 PRINT AT 21,0;"MISSED"
 785 LET A$(P,I,J)="."
 786 LET A$(P+2,I,J)=" "
 787 PRINT AT J+2,3+14*(P-1)+I;"
"
 790 PAUSE 80
 800 LET K=P
 810 GOTO 740
 900 FOR X=W TO W+1
 910 FOR Y=1 TO 10
 920 FOR Z=1 TO 10
 930 LET A$(X,Y,Z)=W$
 940 NEXT Z
 950 NEXT Y
 960 NEXT X
 970 RETURN
1000 PRINT "     10";
1010 GOTO 400
1020 PRINT "   10";
1030 GOTO 570
1060 PRINT AT 21,0;"NOT ALLOWED
"
```

```
1070 PAUSE 50
1090 GOTO 207
1100 DIM B$(6,19)
1110 LET B$(1)="109DESTROYER"
1120 LET B$(2)="209SUDMARINE"
1130 LET B$(3)="316AIRCRAFT CARR
IER"
1140 LET B$(4)="412TORPEDO BOAT"
1150 LET B$(5)="507CRUISER"
1160 LET B$(6)="610BATTLESHIP"
1170 RETURN
1200 IF M=1 THEN GOTO 1300
1210 PRINT AT 21,0;"I BEG YOUR P
ARDON?            ";AT 21,19;
1220 GOTO 2000
1300 PRINT AT 21,0;"I BEG YOUR P
ARDON?"
1310 PAUSE 50
1320 PRINT AT 19,15;"   ";AT 21,
0;"                     ";AT 19,
15;
1330 GOTO 2000
1450 PRINT AT 21,0;"TRIED ALREAD
Y";
1460 IF A$(P,I,J)<>"." THEN PRIN
T " -RUINS LEFT";
1470 PAUSE 50
1475 PRINT AT 21,0;"
              "

1480 GOTO 750
1500 IF A$(P,I,J)="*" OR A$(P,I,
J)="." THEN GOTO 1450
1510 LET A$(P+2,I,J)=A$(P,I,J)
1515 PRINT AT J+2,3+14*(P-1)+I;A
$(P,I,J)
1520 LET V=1
1530 IF B$(V,4)=A$(P,I,J) THEN G
OTO 1560
1540 LET V=V+1
1550 GOTO 1530
1560 PRINT AT 21,0;"HIT A ";B$(V
,4 TO 19);
1570 PRINT " SCORE ";7-V
1575 PAUSE 120
1580 LET S(K)=S(K)+7-V
1582 LET A$(P,I,J)="*"
1583 PRINT AT 17,10+(K-1)*17;S(K
)
1585 IF S(K)=H/2 THEN GOTO 1600
1590 GOTO 743
1600 CLS
1610 PRINT
1620 PRINT "TOTAL FLEET DESTROYE
D"
1630 PRINT
1640 LET V=S(K)-S(P)
```

```
1650 PRINT O$(K);" IS THE WINNER
 BY ";V;" POINTS"
1660 STOP
1700 IF A$(X,I,J)<>" " OR I<1 OR
 I>10 OR J<1 OR J>10 THEN GOTO 1
060
1710 LET G=0
1720 LET C=I-1
1730 LET E=J-1
1740 IF I=1 THEN LET C=1
1750 IF J=1 THEN LET E=1
1760 LET D=I+1
1770 LET F=J+1
1780 IF I=10 THEN LET D=10
1790 IF J=10 THEN LET F=10
1800 IF A$(X,C,E)=" " THEN GOTO
1830
1810 IF A$(X,C,E)<>B$(R,4) THEN
GOTO 1060
1820 LET G=1
1830 IF C<D THEN GOTO 1920
1840 IF E<F THEN GOTO 1960
1850 IF U>1 AND G=0 THEN GOTO 10
60
1860 GOTO 230
1920 LET C=C+1
1930 GOTO 1800
1960 LET E=E+1
1970 LET C=I-1
1980 IF I=1 THEN LET C=1
1990 GOTO 1800
2000 INPUT F$
2010 PRINT F$
2015 IF LEN F$<2 OR LEN F$>3 THE
N GOTO 1200
2017 IF CODE F$(1)<38 OR CODE F$
(1)>47 OR CODE F$(2)<29 OR CODE
F$(2)>37 THEN GOTO 1200
2020 LET I=CODE F$(1)-37
2030 IF LEN F$=3 THEN GOTO 2060
2040 LET J=VAL F$(2)
2050 RETURN
2060 IF F$(3)<>"0" THEN GOTO 120
0
2065 LET J=VAL F$(2 TO 3)
2070 GOTO 2050
2500 GOSUB 500
2510 GOTO 740
```

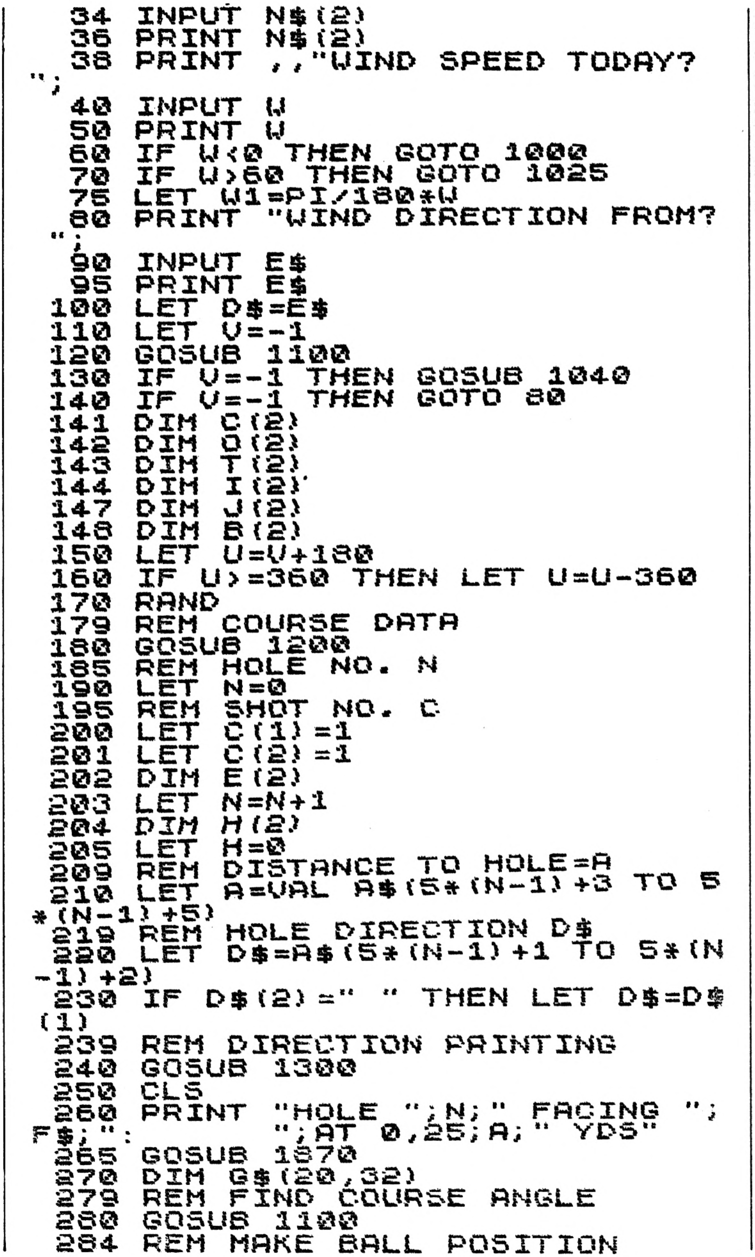
GOLF FOR TWO

Jim Archer began this program as a simple golf routine for one player, then expanded it for two players, adding an enlarged green and a few bunkers. The wind speed and direction are taken into account, and a random element of up to plus or minus 15 per cent keeps the playing down to human standards.

The ZX81 decides which of the two players will go first, and each player in turn then shoots with a certain strength. This is related approximately to the number of yards to the hole, keeping in mind that one square equals ten yards on the main course, one square equals one yard on the green, and at an angle between -180° and +180°. The new ball position is calculated, adding the wind effect and the random element, and is printed at the nearest square as the initial of the player.

If the player lands on the bunker, he has a three in four chance of clearing it. A random square is selected within the sector of the green when landed on, and a satisfying view of the ball disappearing down the hole is shown when a successful shot is made. At the end of an eight hole course the players are graded; this should cause a few chuckles and the desire to tackle another strenuous round.

```
  10 REM GAME OF GOLF FOR TWO
  15 DIM N$(2,3)
  20 PRINT "    ***GAME OF GOLF F
OR 2***","  THIS IS AN EIGHT HOL
E COURSE"
  22 PRINT
  24 PRINT "NAME OF 1ST PLAYER(3
 LETS)? ";
  26 INPUT N$(1)
  28 PRINT N$(1)
  30 PRINT "NAME OF 2ND PLAYER(3
 LETS)? ";
```

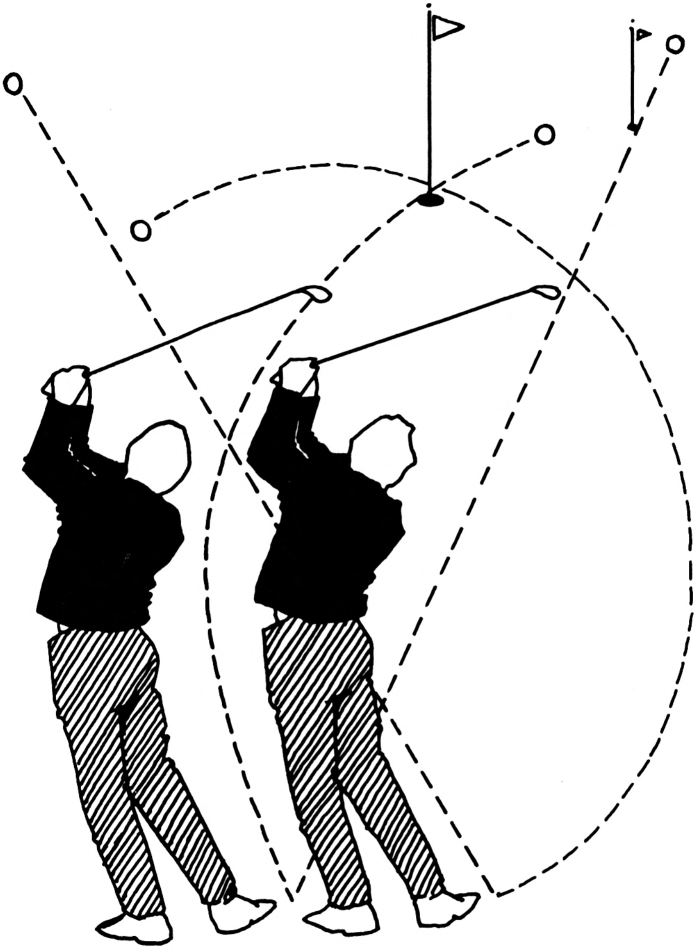


```
  34 INPUT N$(2)
  36 PRINT N$(2)
  38 PRINT ,,"WIND SPEED TODAY?
";
  40 INPUT W
  50 PRINT W
  60 IF W<0 THEN GOTO 1000
  70 IF W>60 THEN GOTO 1025
  75 LET W1=PI/180*W
  80 PRINT "WIND DIRECTION FROM?
";
  90 INPUT E$
  95 PRINT E$
 100 LET D$=E$
 110 LET V=-1
 120 GOSUB 1100
 130 IF V=-1 THEN GOSUB 1040
 140 IF V=-1 THEN GOTO 80
 141 DIM C(2)
 142 DIM O(2)
 143 DIM T(2)
 144 DIM I(2)
 147 DIM J(2)
 148 DIM B(2)
 150 LET U=V+180
 160 IF U>=360 THEN LET U=U-360
 170 RAND
 179 REM COURSE DATA
 180 GOSUB 1200
 185 REM HOLE NO. N
 190 LET N=0
 195 REM SHOT NO. C
 200 LET C(1)=1
 201 LET C(2)=1
 202 DIM E(2)
 203 LET N=N+1
 204 DIM H(2)
 205 LET H=0
 209 REM DISTANCE TO HOLE=A
 210 LET A=VAL A$(5*(N-1)+3 TO 5
*(N-1)+5)
 219 REM HOLE DIRECTION D$
 220 LET D$=A$(5*(N-1)+1 TO 5*(N
-1)+2)
 230 IF D$(2)=" " THEN LET D$=D$
(1)
 239 REM DIRECTION PRINTING
 240 GOSUB 1300
 250 CLS
 260 PRINT "HOLE ";N;" FACING ";
F$;"          ";AT 0,25;A;" YDS"
 265 GOSUB 1870
 270 DIM G$(20,32)
 279 REM FIND COURSE ANGLE
 280 GOSUB 1100
 284 REM MAKE BALL POSITION
 285 LET PO=V*4/45
 290 LET I=VAL E$(PO+1 TO PO+2)
 295 LET J=VAL E$(PO+3 TO PO+4)
 300 DIM G$(20,31)
 301 REM CHOOSE RANDOM PLAYER
 302 LET M=INT (RND*2)+1
 310 LET G$(I,J)=N$(M,1)
 320 PRINT AT I,J;N$(M,1)
 325 LET V1=V*PI/180
 330 LET X=I-INT (A/10*COS V1+.5
)
 340 LET Y=J+INT (A/10*SIN V1+.5
)
 345 LET G$(X,Y)="▩"
 350 PRINT AT X,Y;"▩"
 359 REM PRINT GREEN
 360 FOR R=X-1 TO X+1
 370 FOR S=Y-1 TO Y+1
 380 IF NOT (R=X AND S=Y) THEN G
OSUB 1400
 390 NEXT S
 400 NEXT R
 402 LET P=3-M
 406 LET I(1)=I
 407 LET I(2)=I
 408 LET J(1)=J
 409 LET J(2)=J
 410 GOSUB 1700
 411 IF C(M)>1 THEN PRINT AT 0,2
5;"SHOT ";C(M);" "
 412 GOSUB 415
 414 GOTO 510
 415 PRINT AT 21,0;N$(M);" SHOOT
 WITH STRENGTH?        ";AT 21,26;
 420 INPUT K
 430 PRINT K
 440 PAUSE 50
 450 PRINT AT 21,0;"AT AN ANGLE
OF?                ";AT 21,18;
 460 INPUT T
 470 PRINT T;" DEGREES"
 475 LET T(M)=T(M)+1
 475 PRINT AT M+1,30;T(M)
 480 IF E(M)=1 THEN GOSUB 1830
 490 LET K=K+.4*(RND-.5)*K
 500 LET T=T+30*(RND-.5)
 505 RETURN
 509 REM WIND EFFECT
 510 LET Q=.004*K*W
 515 IF E(M)=1 THEN GOTO 585
 520 PRINT AT I(M),J(M);" "
 525 LET G$(I(M),J(M))=" "
 528 IF C(P)=1 THEN PRINT AT I,J
;N$(P,1)
 550 LET T1=T*PI/180
 560 LET I(M)=I(M)-INT ((K*COS T
1+Q*COS W1)/10)
 570 LET J(M)=J(M)+INT ((K*SIN T
1+Q*SIN W1)/10)
 574 REM SEE IF ON TOP
 575 IF I(M)=I(P) AND J(M)=J(P)
THEN LET J(M)=J(M)+1
 579 REM CHECK INSIDE BORDER
 580 GOSUB 1430
 584 REM SEE IF IN BUNKER
 585 IF G$(I(M),J(M))="▓" THEN G
OSUB 1800
 590 PRINT AT I(M),J(M);N$(M,1)
 595 LET G$(I(M),J(M))=N$(M,1)
 605 LET C(M)=C(M)+1
 610 IF ABS (X-I(M))<2 AND ABS (
Y-J(M))<2 THEN GOSUB 2200
 611 IF C(P)>0 THEN GOSUB 1260
 612 IF H(M)=1 THEN GOTO 620
 615 GOTO 411
 620 CLS
 630 PRINT "HOLE ";N;" :ON THE G
REEN:"
 635 FAST
 636 GOSUB 1870
 637 IF H=1 THEN GOTO 2495
 640 DIM H$(9,9)
 645 LET H=1
 650 FOR R=1 TO 9
 660 FOR S=1 TO 9
 675 LET H$(R,S)="▒"
 680 IF R=5 AND S=5 THEN LET H$(
R,S)="▓"
 690 PRINT AT R+2,S+9;H$(R,S)
 700 NEXT S
 710 NEXT R
 742 SLOW
 745 LET H$(I(M),J(M))=N$(M,1)
 750 PRINT AT I(M)+2,J(M)+9;N$(M
,1)
 755 IF H(P)=1 AND C(P)>0 THEN P
RINT AT I(P)+2,J(P)+9;N$(P,1)
 760 IF I(M)=5 AND J(M)=5 THEN G
OTO 1590
 765 PRINT AT 0,24;"SHOT ";C(M)
 770 GOSUB 415
 780 PRINT AT I(M)+2,J(M)+9;"▒"
 790 LET H$(I(M),J(M))="▒"
 800 LET T1=T*PI/180
 810 LET I(M)=I(M)-INT (K*COS T1
+.5)
 820 LET J(M)=J(M)+INT (K*SIN T1
+.5)
 830 IF H(M)=H(P) THEN GOSUB 186
0
 839 REM CHECK ON GREEN
 840 GOSUB 1480
 845 IF C(P)>0 AND I(M)=I(P) AND
 J(M)=J(P) THEN LET J(M)=J(M)+1
 850 PRINT AT I(M)+2,J(M)+9;N$(M
,1)
 860 IF I(M)=5 AND J(M)=5 THEN G
OTO 1600
 870 LET H$(I(M),J(M))=N$(M,1)
 880 PRINT AT I(M)+2,J(M)+9;N$(M
,1)
 900 LET C(M)=C(M)+1
 910 IF C(P)=0 THEN GOTO 765
 920 GOSUB 1260
 950 IF H(M)=1 THEN GOTO 765
 980 GOTO 1900
1000 REM FALSE WIND SPEEDS
1010 PRINT "BE MORE POSITIVE"
1020 GOTO 30
1025 PRINT "WE CANT PLAY TODAY,
THERES A"," GALE BLOWING"
1030 STOP
1040 PRINT "I BEG YOUR PARDON?"
1050 RETURN
1100 IF D$="N" THEN LET V=0
1110 IF D$="NE" THEN LET V=45
1120 IF D$="E" THEN LET V=90
1130 IF D$="SE" THEN LET V=135
1140 IF D$="S" THEN LET V=180
1150 IF D$="SW" THEN LET V=225
1160 IF D$="W" THEN LET V=270
1170 IF D$="NW" THEN LET V=315
1180 RETURN
1200 LET A$="NW240W 280N 175SE23
0N 180SE250NE220E 275"
1210 LET E$="2015200311010103011
5012811302028"
1220 LET B$="0413052111251627161
4130109060606"
1250 RETURN
1260 LET M=P
1270 LET P=3-M
1280 RETURN
1300 IF D$="N" THEN LET F$="NORT
H"
1310 IF D$="NE" THEN LET F$="NOR
TH-EAST"
1320 IF D$="E" THEN LET F$="EAST
"
1330 IF D$="SE" THEN LET F$="SOU
TH-EAST"
1335 IF D$="S" THEN LET F$="SOUT
H"
1340 IF D$="SW" THEN LET F$="SOU
TH-WEST"
1350 IF D$="W" THEN LET F$="WEST
"
1360 IF D$="NW" THEN LET F$="NOR
```

```
TH-WEST"
1370 RETURN
1400 LET G$(R,S)="▒"
1410 PRINT AT R,S;"▒"
1420 RETURN
1430 IF I(M)<1 THEN LET I(M)=1
1440 IF I(M)>20 THEN LET I(M)=20
1450 IF J(M)<1 THEN LET J(M)=1
1460 IF J(M)>31 THEN LET J(M)=31
1470 RETURN
1480 IF I(M)<1 THEN LET I(M)=1
1490 IF I(M)>9 THEN LET I(M)=9
1500 IF J(M)<1 THEN LET J(M)=1
1510 IF J(M)>9 THEN LET J(M)=9
1520 RETURN
1590 LET C(M)=C(M)-1
1600 PRINT AT 7,14;"*"
1601 LET G$(X,Y)=N$(M,1)
1620 PRINT AT 21,0;"           ███ HO
LE IN ";C(M);" ███           "
1625 PAUSE 140
1630 LET C(M)=0
1635 PRINT AT 7,14;"█"
1640 IF C(P)>0 THEN GOTO 2400
1660 IF N<8 THEN GOTO 200
1665 FOR M=1 TO 2
1670 CLS
1675 LET TOT=T(M)
1680 PRINT AT 7,10;N$(M);AT 9,6;
"***********";AT 10,7;"TOTAL ";T
(M);AT 11,6;"***********";AT 15,
4;"GRADE ";
1681 IF TOT<=20 THEN PRINT "A -
EXCELLENT"
1682 IF TOT>20 AND TOT<=25 THEN
PRINT "B - GOOD"
1683 IF TOT>25 AND TOT<=30 THEN
PRINT "C - AVERAGE"
1684 IF TOT>30 AND TOT<=35 THEN
PRINT "D - MEDIOCRE"
1685 IF TOT>35 AND TOT<=40 THEN
PRINT "E - NEEDS IMPROVING"
1686 IF TOT>40 THEN PRINT "F - M
ISERABLE FAILURE"
1687 STOP
1689 NEXT M
1690 STOP
1700 LET A=VAL B$(PO+1 TO PO+2)
1710 LET B=VAL B$(PO+3 TO PO+4)
1720 IF D$="N" OR D$="S" THEN GO
TO 1780
1730 FOR D=A TO A+2
1740 LET G$(D,B)="█"
1750 PRINT AT D,B;"█"
1760 NEXT D
1770 RETURN
```

```
1780 LET G$(A,B TO B+2)="▒▒▒"
1790 PRINT AT A,B;"▒▒▒"
1795 RETURN
1800 PRINT AT 21,0;" ▒▒▒ IN THE
BUNKER ▒▒▒
1805 PAUSE 90
1810 LET B(M)=1
1820 RETURN
1830 IF RND<.25 THEN RETURN
1840 LET B(M)=0
1850 RETURN
1870 PRINT AT 1,27;"TOT";AT 2,26
;N$(1);":";T(1);AT 3,26;N$(2);":
";T(2)
1880 RETURN
1900 FAST
1905 CLS
1906 PRINT "HOLE ";N;" FACING ";
F$
1907 GOSUB 1870
1910 FOR R=X-1 TO X+1
1920 FOR S=Y-1 TO Y+1
1930 PRINT AT R,S;"▒"
1940 NEXT S
1950 NEXT R
1952 PRINT AT X,Y;"▣"
1955 PRINT AT I(M),J(M);N$(M,1)
1960 SLOW
1970 GOTO 410
2000 DIM H(2)
2010 PRINT H(1),H(2)
2020 LET H(1)=1
2030 PRINT H(1),H(2)
2200 LET I(M)=4+3*(I(M)-X)+INT (
RND*10/3.34)
2210 LET J(M)=4+3*(J(M)-Y)+INT (
RND*10/3.34)
2220 LET H(M)=1
2225 IF H(P)=1 AND I(M)=I(P) AND
 J(M)=J(P) THEN LET J(M)=J(M)+1
2230 RETURN
2400 GOSUB 1260
2410 IF H(M)=0 THEN GOTO 1900
2420 GOTO 760
2496 CLS
2497 PRINT "HOLE ";N;" :ON THE G
REEN:"
2498 GOSUB 1870
2505 FAST
2510 FOR R=1 TO 9
2520 FOR S=1 TO 9
2530 IF R=5 AND S=5 THEN LET H$(
R,S)="▣"
2540 PRINT AT R+2,S+9;H$(R,S)
2550 NEXT S
```

```
2560 NEXT R
2570 GOTO 742
```



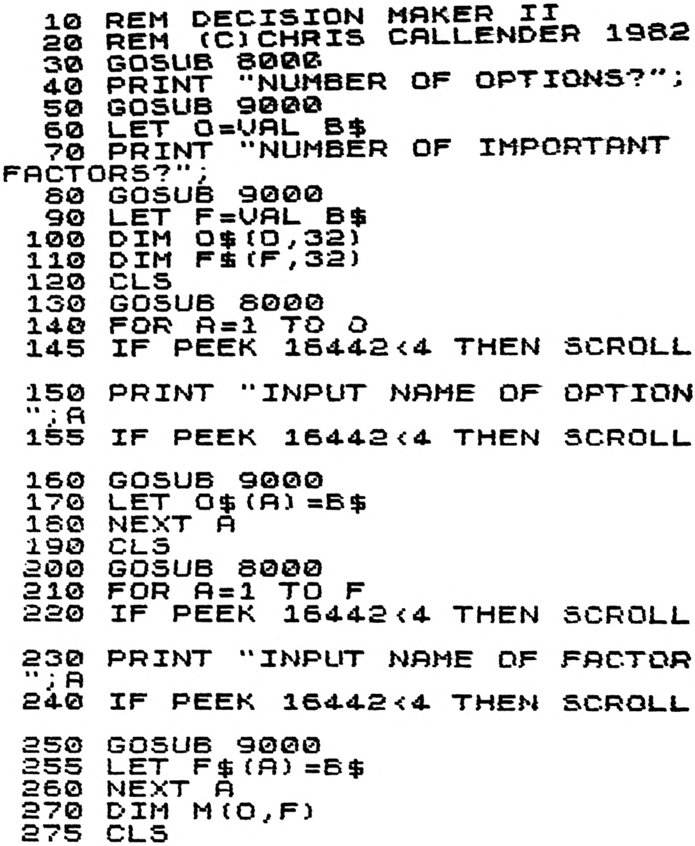
THE ZX81 LENDS A HAND

DECISION MAKER

Chris Callender's Decision Maker is the program for anyone who has trouble making up their mind. It may well act as a genuine aid in making balanced judgements. Here is a sample of how it works. From the sample run, you will see how to use the program to solve your own problems. This run attempts to aid the decision of which pet to buy.

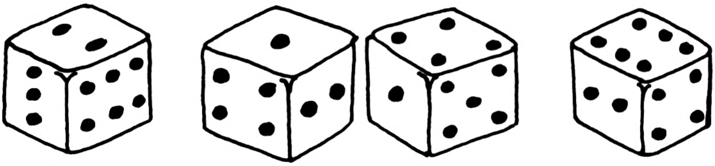
```
NUMBER OF OPTIONS?  3
NUMBER OF IMPORTANT FACTORS?  3
DECISION MAKER II
INPUT NAME OF OPTION  1
PIG
INPUT NAME OF OPTION  2
SWAN
INPUT NAME OF OPTION  3
HAMSTER

DECISION MAKER II
INPUT NAME OF FACTOR  1
COST
INPUT NAME OF FACTOR  2
AFFECTION
INPUT NAME OF FACTOR  3
COLOUR

DECISION MAKER II
HOW WOULD YOU RATE COST ON THE PIG?
55

HOW WOULD YOU RATE COST ON THE
SWAN?
100
```

```
...etc

HOW WOULD YOU RATE AFFECTION ON THE
PIG?
99

...etc

THANKS...HERE IS WHAT I THINK
PIG   326
SWAN   293
HAMSTER   162
```



```
  10 REM DECISION MAKER II
  20 REM (C)CHRIS CALLENDER 1982
  30 GOSUB 8000
  40 PRINT "NUMBER OF OPTIONS?";
  50 GOSUB 9000
  60 LET O=VAL B$
  70 PRINT "NUMBER OF IMPORTANT
FACTORS?";
  80 GOSUB 9000
  90 LET F=VAL B$
 100 DIM O$(O,32)
 110 DIM F$(F,32)
 120 CLS
 130 GOSUB 8000
 140 FOR A=1 TO O
 145 IF PEEK 16442<4 THEN SCROLL

 150 PRINT "INPUT NAME OF OPTION
";A
 155 IF PEEK 16442<4 THEN SCROLL

 160 GOSUB 9000
 170 LET O$(A)=B$
 180 NEXT A
 190 CLS
 200 GOSUB 8000
 210 FOR A=1 TO F
 220 IF PEEK 16442<4 THEN SCROLL

 230 PRINT "INPUT NAME OF FACTOR
";A
 240 IF PEEK 16442<4 THEN SCROLL

 250 GOSUB 9000
 255 LET F$(A)=B$
 260 NEXT A
 270 DIM M(O,F)
 275 CLS
```

```
  277 GOSUB 8000
  280 FOR A=1 TO O
  290 FOR B=1 TO F
  295 CLS
  297 GOSUB 8000
  300 PRINT "HOW WOULD YOU RATE "
  F$(B)
  310 PRINT "ON THE ";O$(A)
  320 GOSUB 9000
  330 LET M(A,B)=VAL B$
  340 NEXT B
  350 NEXT A
  360 CLS
  370 GOSUB 8000
  380 PRINT "THANKS...HERE IS WHA
T I THINK"
  390 FOR A=1 TO O
  400 LET C=0
  410 FOR B=1 TO F
  420 LET C=C+M(A,B)
  430 NEXT B
  440 PRINT O$(A);C
  450 NEXT A
  460 PRINT "ANOTHER RUN?";
  470 GOSUB 9000
  480 CLS
  490 IF B$="YES" THEN RUN
  500 STOP
 8000 PRINT TAB 5;"DECISION MAKER
  II"
 8010 PRINT TAB 5;"██████████████
████"
 8020 RETURN
 9000 REM INPUT ROUTINE
 9010 LET B$=""
 9020 PAUSE 4E4
 9022 LET A$=INKEY$
 9025 IF A$=CHR$ 118 THEN GOTO 90
60
 9027 IF INKEY$=CHR$ 115 THEN LET
  A$=" "
 9030 PRINT A$;
 9040 LET B$=B$+A$
 9050 GOTO 9020
 9060 PRINT
 9070 RETURN
```



AUDIO MORSE TEST

This program, written by Chris Callender, chooses a word and plays it (in Morse code) through your television loudspeaker, using machine code to make the noise. The first program sets up the machine code in a REM statement in line 10. Once you have RUN the first program, enter LIST and you will see the characters after the REM have been changed. If you want to use this routine in your own programs, key in the first program, then delete lines 20 to 50. To make the sound, you use:

```
POKE 16523, duration
LET L USR 16514
```

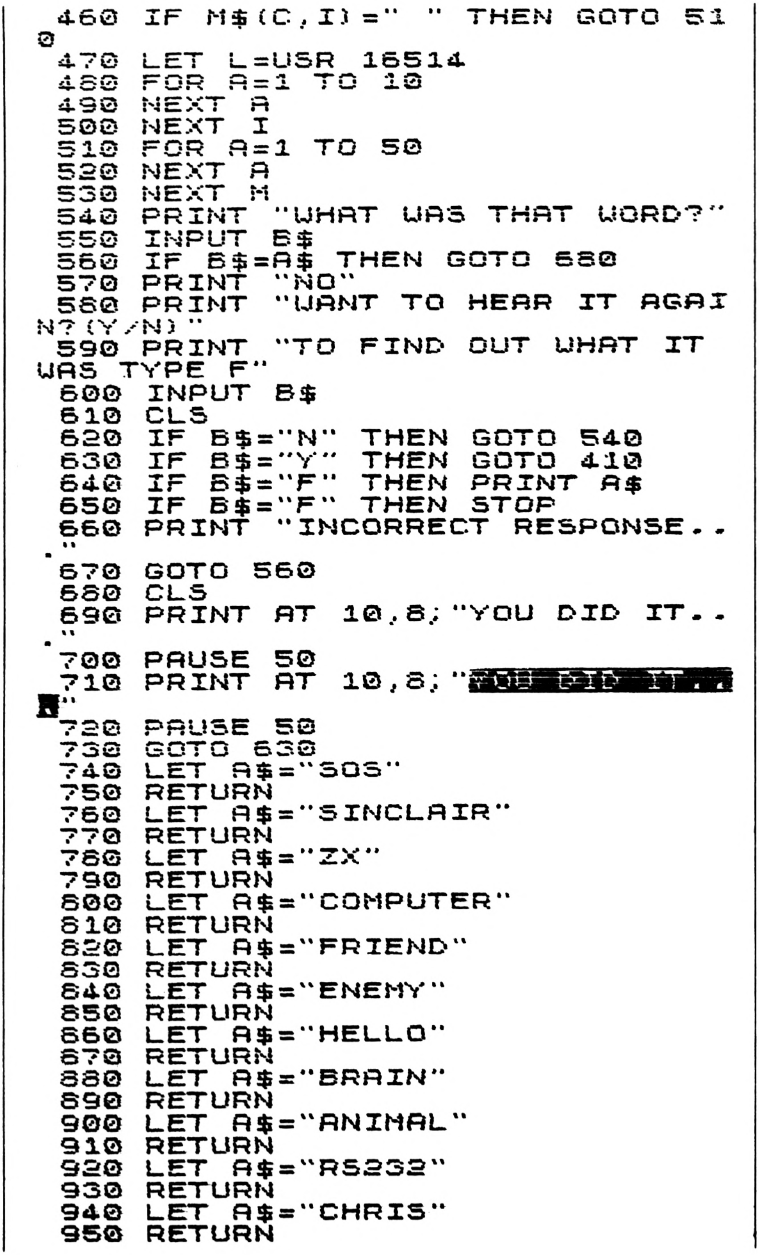
Enter the second program to get the Morse test. By changing the 30 in line 390 to the number of words you want, you could add more words after line 1350. The machine code is stored in line 10. Here is a disassembly:

```
LD A,00
EXX
CALL 0320
EXX
INC A
CP duration
JP nz,4084
RET
```

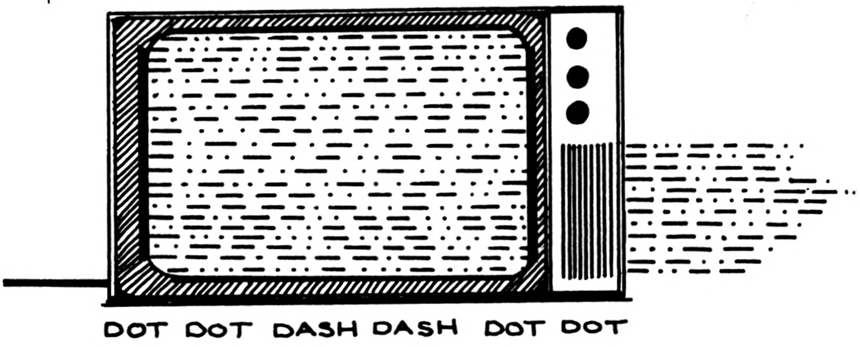
Lines 20 to 380 store the Morse code in array M$, and line 390 selects a word at random. Lines 510 to 530 play the Morse code sounds, and lines 540 to 670 ask you what the word was. Finally, here is a message for you to decode:

```
....  .-...-.  ..-.  ..--.
```

```
  10 REM 12345678901234
  20 LET H$="3E0008CD2003083CFE0
0C28440C9"
  30 FOR A=1 TO 26 STEP 2
  40 POKE 16514+INT (A/2),(CODE
A$(A)-28)*16+(CODE A$(A+1)-28)
  50 NEXT A

RUN
DELETE LINES 20 50
THEN TYPE IN:-
   20 DIM M$(36,5)
   30 LET M$(1)="-----"
   40 LET M$(2)=".----"
   50 LET M$(3)="..---"
   60 LET M$(4)="...--"
   70 LET M$(5)="....-"
   80 LET M$(6)="....."
   90 LET M$(7)="-...."
  100 LET M$(8)="--..."
  110 LET M$(9)="---.."
  120 LET M$(10)="----."
  130 LET M$(11)=".-"
  140 LET M$(12)="-..."
  150 LET M$(13)="-.-."
  160 LET M$(14)="-.."
  170 LET M$(15)="."
  180 LET M$(16)="..-."
  190 LET M$(17)="--."
  200 LET M$(18)="...."
  210 LET M$(19)=".."
  220 LET M$(20)=".---"
  230 LET M$(21)="-.-"
  240 LET M$(22)=".-.."
  250 LET M$(23)="--"
  260 LET M$(24)="-."
  270 LET M$(25)="---"
  280 LET M$(26)=".--."
  290 LET M$(27)="--.-"
  300 LET M$(28)=".-."
  310 LET M$(29)="..."
  320 LET M$(30)="-"
  330 LET M$(31)="..-"
  340 LET M$(32)="...-"
  350 LET M$(33)=".--"
  360 LET M$(34)="-..-"
  370 LET M$(35)="-.--"
  380 LET M$(36)="--.."
  390 LET N=INT (RND*30)
  400 GOSUB 740+N*20
  410 FOR M=1 TO LEN A$
  420 LET C=CODE A$(M)-27
  430 FOR I=1 TO 5
  440 IF M$(C,I)="." THEN POKE 16
523,10
  450 IF M$(C,I)="-" THEN POKE 16
523,40
```



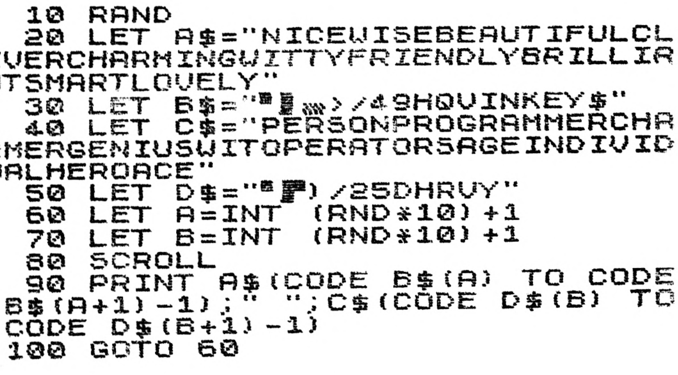
```
 460 IF M$(C,I)=" " THEN GOTO 51
0
 470 LET L=USR 16514
 480 FOR A=1 TO 10
 490 NEXT A
 500 NEXT I
 510 FOR A=1 TO 50
 520 NEXT A
 530 NEXT M
 540 PRINT "WHAT WAS THAT WORD?"
 550 INPUT B$
 560 IF B$=A$ THEN GOTO 680
 570 PRINT "NO"
 580 PRINT "WANT TO HEAR IT AGAI
N?(Y/N)"
 590 PRINT "TO FIND OUT WHAT IT
WAS TYPE F"
 600 INPUT B$
 610 CLS
 620 IF B$="N" THEN GOTO 540
 630 IF B$="Y" THEN GOTO 410
 640 IF B$="F" THEN PRINT A$
 650 IF B$="F" THEN STOP
 660 PRINT "INCORRECT RESPONSE..
."
 670 GOTO 560
 680 CLS
 690 PRINT AT 10,8;"YOU DID IT..
."
 700 PAUSE 50
 710 PRINT AT 10,8;"YOU DID IT..
."
 720 PAUSE 50
 730 GOTO 630
 740 LET A$="SOS"
 750 RETURN
 760 LET A$="SINCLAIR"
 770 RETURN
 780 LET A$="ZX"
 790 RETURN
 800 LET A$="COMPUTER"
 810 RETURN
 820 LET A$="FRIEND"
 830 RETURN
 840 LET A$="ENEMY"
 850 RETURN
 860 LET A$="HELLO"
 870 RETURN
 880 LET A$="BRAIN"
 890 RETURN
 900 LET A$="ANIMAL"
 910 RETURN
 920 LET A$="RS232"
 930 RETURN
 940 LET A$="CHRIS"
 950 RETURN
 960 LET A$="MORSE"
 970 RETURN
 980 LET A$="HEX"
 990 RETURN
1000 LET A$="HORROR"
1010 RETURN
1020 LET A$="MICRO"
1030 RETURN
1040 LET A$="PRINTER"
1050 RETURN
1060 LET A$="RAINBOW"
1070 RETURN
1080 LET A$="FORTUNE"
1090 RETURN
1100 LET A$="ZX99"
1110 RETURN
1120 LET A$="CASSETTE"
1130 RETURN
1140 LET A$="16KRAM"
1150 RETURN
1160 LET A$="GLASS"
1170 RETURN
1180 LET A$="CAR"
1190 RETURN
1200 LET A$="HELP"
1210 RETURN
1220 LET A$="PEN"
1230 RETURN
1240 LET A$="KING"
1250 RETURN
1260 LET A$="QUEEN"
1270 RETURN
1280 LET A$="RADIO"
1290 RETURN
1300 LET A$="DOOR"
1310 RETURN
1320 LET A$="KEY"
1330 RETURN
1340 LET A$="GOOD"
1350 RETURN
```



COMPLIMENT GENERATOR

When you are feeling tired and miserable after working all night at a program (and it still doesn't work), you can cheer yourself up with this 1K program, written by Dilwyn Jones, which will generate a complimentary phrase every time you press a key. Using this method, you can use the program to generate insults (as the next program illustrates), buzzwords, or character names for your next adventure program. The graphics characters in B$ are shift 1, shift 5, shift D and the final character is the function INKEY$. The graphics characters in D$ are shift 1 and shift E.

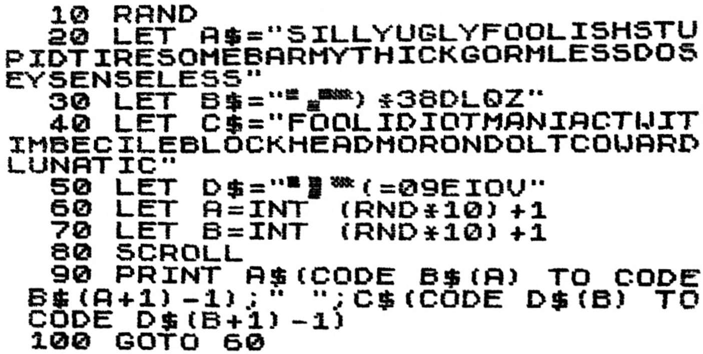
All the words used in this program are held in the two strings A$ and C$. The strings B$ and D$ contain pointers whose CODE is where each word starts in A$ and C$ respectively. A is the variable that determines which word is to be extracted from A$, and C is the variable which aids in the selection of the word from C$. Line 80 keeps the display scrolling upwards neatly, so that each word appears below the previous one. All the decoding and printing is done in line 90 and line 100 keeps the compliments coming.



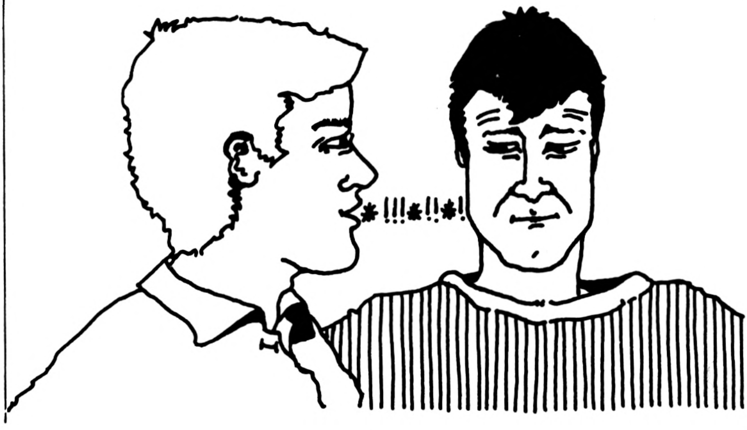
```
  10 RAND
  20 LET A$="NICEWISEBEAUTIFULCL
EVERCHARMINGWITTYFRIENDLYBRILLIA
NTSMARTLOVELY"
  30 LET B$="▘▐▒>/49HQVINKEY$"
  40 LET C$="PERSONPROGRAMMERCHA
RMERGENIUSWITOPERATORSAGEINDIVID
UALHEROACE"
  50 LET D$="▘▛)/25DHRVY"
  60 LET A=INT (RND*10)+1
  70 LET B=INT (RND*10)+1
  80 SCROLL
  90 PRINT A$(CODE B$(A) TO CODE
 B$(A+1)-1);" ";C$(CODE D$(B) TO
 CODE D$(B+1)-1)
 100 GOTO 60
```

INSULT GENERATOR

This program, also from Dilwyn Jones, is a variation on the previous one. Although it uses no swear-words, some of the insults are fairly fierce, so don't try it out on your best friends. The graphics characters in B$ are shift 1, shift T and shift S, while those in D$ are shift 1, shift 5 and shift S.

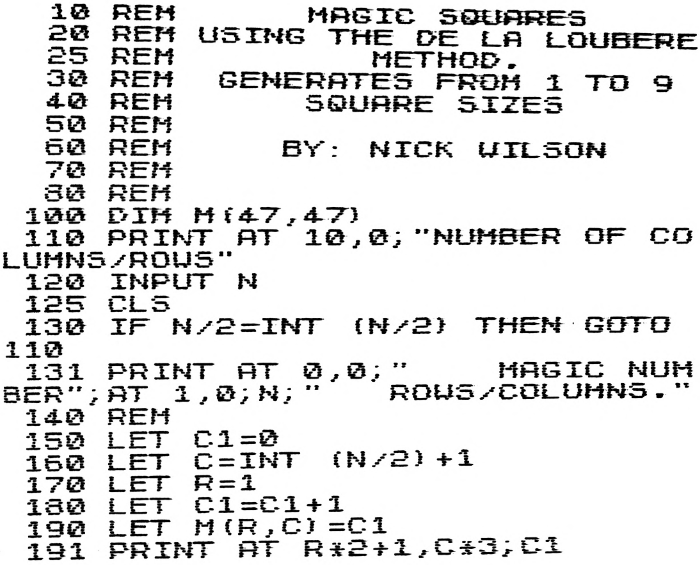


```
   10 RAND
   20 LET A$="SILLYUGLYFOOLISHSTU
PIDTIRESOMEBARMYTHICKGORMLESSDOS
EYSENSELESS"
   30 LET B$="▘▞▒)*38DLQZ"
   40 LET C$="FOOLIDIOTMANIACTWIT
IMBECILEBLOCKHEADMORONDOLTCOWARD
LUNATIC"
   50 LET D$="▘▌▒(=09EIOV"
   60 LET A=INT (RND*10)+1
   70 LET B=INT (RND*10)+1
   80 SCROLL
   90 PRINT A$(CODE B$(A) TO CODE
 B$(A+1)-1);" ";C$(CODE D$(B) TO
 CODE D$(B+1)-1)
  100 GOTO 60
```

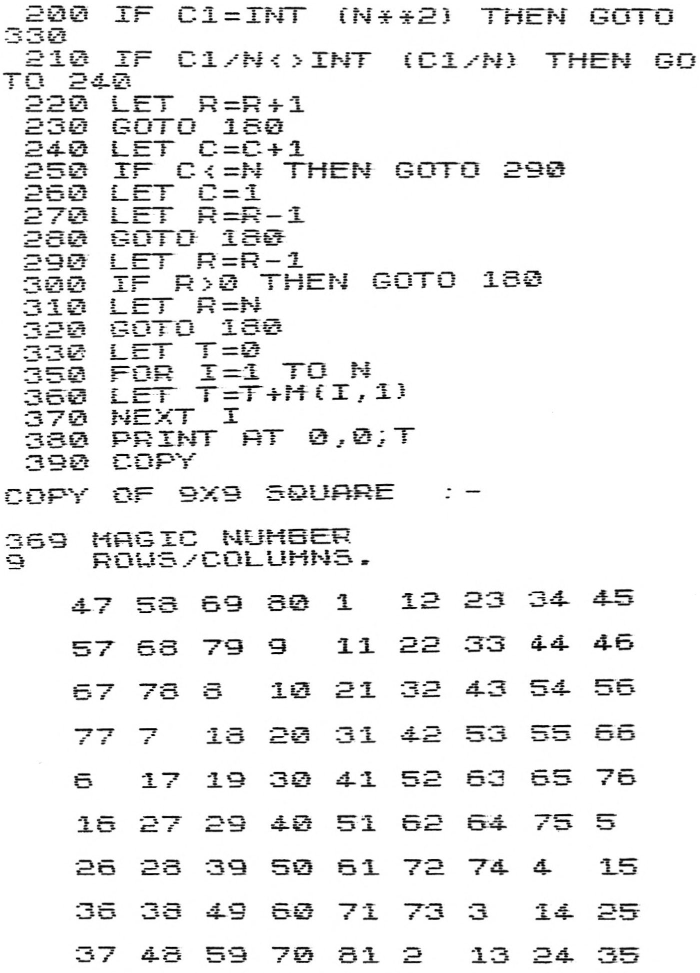


LOUBERE MAGIC SQUARES

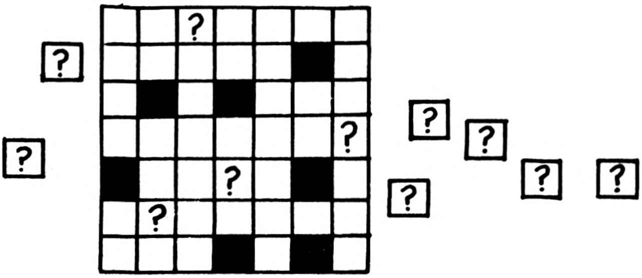
This program, by Nick Wilson, will generate four different sizes of magic squares, using the de la Loubere method. A 'magic square' is a square of numbers where all the rows, columns and leading diagonals individually total the same number. When you run the program you will be asked to enter an integer between one and nine. Only odd numbers are accepted, and your input is checked 'for oddness' in line 130. Next the screen clears, a magic square is constructed, then COPYed to the printer.



```
  10 REM        MAGIC SQUARES
  20 REM USING THE DE LA LOUBERE
  25 REM          METHOD.
  30 REM  GENERATES FROM 1 TO 9
  40 REM        SQUARE SIZES
  50 REM
  60 REM       BY: NICK WILSON
  70 REM
  80 REM
 100 DIM M(47,47)
 110 PRINT AT 10,0;"NUMBER OF CO
LUMNS/ROWS"
 120 INPUT N
 125 CLS
 130 IF N/2=INT (N/2) THEN GOTO
110
 131 PRINT AT 0,0;"     MAGIC NUM
BER";AT 1,0;N;"    ROWS/COLUMNS."
 140 REM
 150 LET C1=0
 160 LET C=INT (N/2)+1
 170 LET R=1
 180 LET C1=C1+1
 190 LET M(R,C)=C1
 191 PRINT AT R*2+1,C*3;C1
```



```
 200 IF C1=INT (N**2) THEN GOTO
330
 210 IF C1/N<>INT (C1/N) THEN GO
TO 240
 220 LET R=R+1
 230 GOTO 180
 240 LET C=C+1
 250 IF C<=N THEN GOTO 290
 260 LET C=1
 270 LET R=R-1
 280 GOTO 180
 290 LET R=R-1
 300 IF R>0 THEN GOTO 180
 310 LET R=N
 320 GOTO 180
 330 LET T=0
 350 FOR I=1 TO N
 360 LET T=T+M(I,1)
 370 NEXT I
 380 PRINT AT 0,0;T
 390 COPY
COPY OF 9X9 SQUARE  :-

369 MAGIC NUMBER
9   ROWS/COLUMNS.

   47 58 69 80 1  12 23 34 45

   57 68 79 9  11 22 33 44 46

   67 78 8  10 21 32 43 54 56

   77 7  18 20 31 42 53 55 66

   6  17 19 30 41 52 63 65 76

   16 27 29 40 51 62 64 75 5

   26 28 39 50 61 72 74 4  15

   36 38 49 60 71 73 3  14 25

   37 48 59 70 81 2  13 24 35
```



How To Write Better Programs

By Tim Hartnell, series editor

There are a number of fine programs in this book, and many of the regular computer magazines contain other such ones. But no matter how good the programs from published sources are, you are certain to get more pleasure from running them if they have been partially or completely written by you. Putting your personal stamp on programs, altering them to reflect your wishes and creativity, is an excellent way to improve the programs, and eventually, of course, you'll become a better and more imaginative programmer.

Programs in magazines, and in books like this one, are ideal as starting points for your own developments. You may also find that advertisements for software packages can be fruitful 'idea-starters'. You only need to read the description of what the commercially available program does, and you will have the first step towards creating your own program. You have to be careful, of course, not to infringe copyright either in the screen displays, in the name of the program, or the names of the 'characters' within the program. However, you will probably find that at a certain point in its development the program will take on a life of its own, growing and evolving away from the original scenario, until you eventually have a completely new game concept and implementation.

Whatever you do, be careful not to pass off other people's work as your own. By all means adapt and improve published programs, but do not then present them to magazines as if they were originals. I have lost count of the number of times one of my own programs, from one of my books, has been submitted to me for publication.

Always watch out for new ideas as you look through books, game and computer magazines, or wander through video game arcades. It may be worth keeping notes of ideas you come across for games, for character shapes, for sounds, for dramatic endings and so on. Thus you will never be short of ideas, and you will also be able to merge the material together to produce better games which hold the player's attention for longer.

Games tend to fall into one of three categories, and it is worth making sure of the category into which your proposed program will fall *before* you start to program, since the category of game materially alters the programming approach. This is not to say that, as you develop a program, it will not move from one category into another, nor that a particular game might not extend across two categories, but it is nevertheless useful to keep the various groups separate in your mind, just to clarify your thoughts. The three categories are:

1. Board games
2. 'Arcade' (that is, highly visual, fast moving, noisy, real time) games
3. Games of chance (such as Roulette and Snap).

In board games, the quality of play is more important than lightning-fast response, while the arcade-type programs must be kept moving at all costs, even if some 'intelligence' from your Martian intruders must be sacrificed to achieve this. Games of chance depend more on their ease of play ('user-friendly' inputs), and an approach to true randomness, than do either of the other categories.

You will find that games programs tend to fall into types, which are subdivisions of the three above mentioned categories. Many board games are variants of chess or checkers; many arcade games started off life as Space Invader-type games; and games of chance started off in the 'real world' of dice and cards. Looking at a program description, or a games machine, and trying to categorise the game you see can help trigger new ideas which fit within that particular game's genre.

There is a school of thought within programming — generally called 'structured programming' — which believes that discipline at the beginning of the games-writing process is essential. While less interesting than sitting down at the computer right away, a much better program is produced in the end. I once wrote a program called Dome Dweller, a simulation program in which the player is in charge of a 'lunar dome' and must decide which products to manufacture and sell in order to buy oxygen and food for the station's inhabitants. (This program was used in my book *The Book of Listings*, written with Jeremy Ruston, and published by the BBC.) Once I had decided the overall scenario, I worked out the screen display, and came up with an idea as follows:

```
Oxygen supplies are low
There are 96 people living within your dome in year 3
Money credit is $5,693
Annual maintenance charge is $226
Oxygen tanks hold 811 units
Oxygen costs $8 per unit
Each dome dweller needs 5 units a year
Food stocks stand at 2122
Each dweller needs 3 units a year ($6 each, $576 for dome. This will last 7 years at present population.)
You can trade your unique lunar sculptures with the people who live in other domes. You use up 2 units of oxygen making each one, and sell them for $30.
```

As you can probably guess from this 'sample printout', the idea of the program is to decide how many 'unique lunar sculptures' you must make and sell in order to buy oxygen and food, and to pay the 'annual maintenance' charge. The problem with this particular program is that making each sculpture uses up oxygen, so you must balance your wish to make money against the need to use the oxygen intelligently.

You may well wish to try writing such a program yourself. You should end up with an enjoyable program, and writing it will do much to help you develop your programming skills. The first thing to do is to make a list of what the program has to do:

Set up the needed variables
Tell the player the 'state of the dome'
Ask how much oxygen to be bought
Check if can afford this, if so buy it, if not go back and ask again
Ask how much food to be bought
Check if can afford this, if so buy it, if not go back and ask again
Update oxygen quantity
Update food quantity
Reduce money left total
Ask how many items of sculpture to be made
Check if there is enough oxygen to make this many, if not go back and ask again
Reduce oxygen quantity by amount needed to make the number of sculptures specified, increase money total to reflect value of sculptures made
Increase the population total slightly, add one to the 'current year'
Check if there is enough food in stocks to feed whole population
Check if there is enough oxygen for whole population
Check if there is any money
If any of these conditions are negative (eg not enough food) send action to an 'end of game' routine
If all are positive, loop back to tell the player the state of the dome, and continue to circle

You could probably write a Dome Dweller program using the list above, together with the 'sample printout' information. There is, however, a secret I should like to share with you which unlocks programming problems almost instantly. You can actually write all the vital parts of a program in minutes, so you can see the raw framework of a program like this running long before you fill in the details. And once you have a framework you can work on it for as long as you like, knowing as you do so that — at every moment in program development — you have a working program. You do not have to wait until the end until you can run it to see how you are going. The 'secret' is to hold the entire program within a series of subroutine calls, all held within a perpetual loop. Here's how it could work with this program. The very first lines you enter in your computer are as follows:

```
10 REM DOME DWELLER
20 GOSUB 1000: REM ASSIGN VARIABLES
30 GOSUB 2000: REM PRINT OUT STATE OF
DOME
40 GOSUB 3000: REM OXYGEN
50 GOSUB 4000: REM FOOD
60 GOSUB 5000: REM SCULPTURE
70 GOSUB 6000: REM UPDATE POPULATION
80 GOSUB 7000: REM CHECK ON STATE OF
 DOME
90 IF (all conditions positive, from GOSUB 7000)
THEN GOTO 30
100 REM End of game ...
```

As you can see once you have the 'master loop' set up in this way, it is relatively simple to fill in each of the subroutines one by one, testing each as you do so, and elaborating each one so that you end up eventually with a very good program. The only thing you need now is a list of the variables which you will use with the program.

I find the best way to do this is to use explicit names for variables so that when you are programming you do not have to spend time checking, for example, whether AA stands for the population, or the number of units of oxygen used up in making each item of sculpture. To make programs as easy as possible to transfer between different computers you can stick to two letter variable names, or you can take advantage (if your computer allows it) of long names (such as OXYUSE for the amount of oxygen used) for variables. Then you have no doubts whatsoever as to the meaning of each variable name. To show how this can work, and to illustrate a further advantage of explicit variable names, here are the variables used in Dome Dweller:

FOLK — population of dome
CASH — money in treasury
FOOD — food stocks on hand
FOODCOST — how much each unit of food costs
FOODNEED — how many units of food were consumed per person per year
ARTCOST — how much oxygen was used up making each piece of sculpture
ARTPAY — how many dollars each piece of sculpture was sold for
OXY — oxygen stocks on hand
OXYNEED — how many units of oxygen were consumed per person per year
OXYCOST — how much each unit of oxygen cost to buy
REPAIR — the cost of annual repairs to the dome
YEAR — the year of the dome's life

Using explicit variable names in this way — although they use up more memory than do single or double-letter variable names — makes it very simple to follow through a program, working out what each section of the program actually does. Moreover, and this is the further advantage mentioned, it is very easy when writing the program to insert the formulae required for calculations. By this I mean that if, for example, you wished to include (as I do in this program) an indication of how much oxygen is needed for each year, you simply multiply the number of people in the dome (FOLK) by the number of oxygen units each person needs each year (OXYNEED). You can then include within the printouts for the state of the dome a line like:

```
PRINT "THERE ARE ";FOLK;" IN THE DOME"
PRINT "IN YEAR ";YEAR
PRINT "EACH PERSON NEEDS ";OXYNEED;"
        UNITS OF"
PRINT "OXYGEN EACH YEAR,";
        OXYNEED*FOLK;" NEEDED"
PRINT "FOR THE WHOLE DOME"
```

It also makes it very easy to check on whether purchases are possible. For example, to buy food, you could say:

```
PRINT "HOW MUCH FOOD WILL YOU BUY?"
INPUT A
IF A*FOODCOST >CASH THEN GOTO (get
another A)
```

So the suggestions given here for improving your programs by the use of 'structured programming' include the following:

- draw up a sample printout, or mock-up of the final screen display
- draw up a list of what the program has to do each time through a 'master control loop'
- change this list to a series of subroutine calls
- use explicit variable names if possible

It is useful if you are designing programs for others to use to ensure that it is quite clear what the player should do when running the program. There is little point, especially when memory is limited, in including a long set of instructions within the program, but you should certainly write such instructions down. In addition, user prompts should be explicit (such as ENTER THE NUMBER OF GOES YOU WANT) and should include warnings of the limits which will be placed on the input (HOW MANY CARDS WILL YOU START WITH: 1, 2 OR 3 ?, for instance).

You cannot assume that you will be present every time a program is run, so you should do your best to make it as foolproof as possible. If you can, add error-trapping routines to the program to ensure that a mistake in entering a choice earlier on in the program will not cause it to crash or come up with stupid results later on.

If you read through this section of the book several times and try to apply the ideas to your own programming work, you should find your work quality improves significantly, and also that you can spend more time improving and embellishing a program and less in the raw mechanical task of getting the thing running.

GLOSSARY

A

Accumulator — the place within the computer in which arithmetic computations are performed and where the results of these computations are stored.

Algorithm — the series of steps the computer follows to solve a particular problem.

Alphanumeric — this term is usually used in relation to a keyboard, as in 'it is an alphanumeric keyboard', which means that the keyboard has letters as well as numbers. It is also used to refer to the 'character set' of the computer. The character set comprises the numbers and letters the computer can print on the screen.

ALU (Arithmetic/Logic Unit) — the part of the computer which does arithmetic (such as addition, subtraction) and where decisions are made.

AND — a Boolean logic operation that the computer uses in its decision-making process. It is based on Boolean algebra, a system developed by mathematician George Boole (1815-64). In Boolean algebra the variables of an expression represent a logical operation such as OR and NOR.

ASCII — stands for American Standard Code for Information Exchange, the most widely used encoding system for English language alphanumerics. There are 128 upper and lower case letters, digits and some special characters. ASCII converts the symbols and control instructions into seven-bit binary combinations.

Assembler — a program which converts other programs written in assembly language into machine code (which the computer can understand directly). Assembly language is a low level programming language which uses easily memorised combinations of two or three letters to represent a particular instruction which the assembler then converts so the machine can understand it. Examples of these are ADD (add), and SUB (subtract). A computer programmed in assembly language tends to work more quickly than one programmed in a higher level language such as BASIC.

B

BASIC — an acronym for Beginners All-Purpose Symbolic Instruction Code. It is the most widely used computer language in the microcomputer field. Although it has been criticised by many people, it has the virtue of being very easy to learn. A great number of BASIC statements resemble ordinary English.

Baud — named after Baudot, a pioneer of telegraphic communications. Baud measures the rate of transfer of information and is approximately equal to one bit per second.

BCD — an abbreviation for Binary Coded Decimal.

Benchmark — a test against which certain functions of the computer can be measured. There are a number of so-called 'standard Benchmark tests', but generally these only test speed. This is rarely the aspect of a microcomputer that is most of interest to the potential buyer.

Binary — a numbering system that uses only zeros and ones.

Bit — an abbreviation for Binary Digit. This is the smallest unit of information a computer circuit can recognise.

Boolean Algebra — the system of algebra developed by mathematician George Boole which uses algebraic notation to express logical relationships (see AND).

Bootstrap — a short program or routine which is read into the computer when it is first turned on. It orients the computer to accept the longer, following program.

Bug — an error in a computer program which stops the program from running properly. Although it is generally used to mean only a fault or an error in a program, the term bug can also be used for a fault in the computer hardware.

Bus — a number of conductors used for transmitting signals such as data instructions, or power in and out of a computer.

Byte — a group of binary digits which make up a computer word. Eight is the most usual number of bits in a byte.

C

CAI — Computer Assisted Instruction.

CAL — Computer Assisted Learning. The term is generally used to describe programs which involve the learner with the learning process.

Chip — the general term for the entire circuit which is etched onto a small piece of silicon. The chip is, of course, at the heart of the microcomputer.

Clock — the timing device within the computer that synchronises its operations.

COBOL — a high level language derived from the words Common Business Orientated Language. COBOL is designed primarily for filing and record-keeping.

Comparator — a device which compares two things and produces a signal related to the difference between the two.

Compiler — a computer program that converts high level programming language into binary machine code so the computer can handle it.

Complement — a number which is derived from another according to specified rules.

Computer — a device with three main abilities or functions:
1) to accept data
2) to solve problems
3) to supply results

CPU — stands for Central Processing Unit. This is the heart of the computer's intelligence, where data is handled and instructions are carried out.

Cursor — a character which appears on the TV screen when the computer is operating. It shows where the next character will be printed. On a computer there are usually 'cursor control keys' to allow the user to move the cursor around the screen.

D

Data — information in a form which the computer can process.

Debug — the general term for going through a program and correcting any errors in it, that is, chasing down and removing bugs (see Bug).

Digital Computer —a computer which operates on information which is in a discrete form.

Disk/Disc — this is a magnetically sensitised plastic disk, a little smaller than a single play record. This is used for storing programs and for obtaining data. Disks are considerably faster to load than a cassette of the same length program. The disk can be searched very quickly while a program is running for additional data.

Display — the visual output of the computer, generally on a TV or monitor screen.

Dot Matrix Printer — a printer which prints either the listing of a program or that which is displayed on the TV screen. Each letter and character is made up of a number of dots. The higher the number of dots per character the finer the resolution of the printer.

Dynamic Memory — a memory unit within the computer which 'forgets' its contents when the power is turned off.

E

Editor — this term is generally used for the routine within the computer which allows you to change lines of a program while you are writing it.

EPROM — stands for Erasable Programmable Read-Only Memory. This is like the ROM in the computer, except that it is fairly easy to load material into an EPROM and it doesn't disappear when you turn the power off. EPROMs must be placed in a strong ultra violet light to erase them.

Error Messages — the information given by a computer where there is a fault in the coding during a part of a program, usually shown by the computer stopping, and printing a word, or a word and numbers, or a combination of numbers only, at the bottom of the screen. This tells you what mistake has been made. Common mistakes include using the letter O instead of zero in a line, or leaving out a pair of brackets, or one of the brackets, in an expression, or failing to define a variable.

F

File — a collection of related items of information organised in a systematic way.

Floppy Disk — a relatively cheap form of magnetic disk used for storing computer information, and so named because it is quite flexible (see Disk/Disc).

Flow Chart — a diagram drawn up before writing a program, in which the main operations are enclosed within rectangles or other shapes and connected by lines, with arrows to represent loops, and decisions written at the branches. It makes writing a program much easier because traps such as infinite loops, or non-defined variables can be caught at an early stage. It may not be worth writing a flow chart for very short programs, but generally a flow chart aids in creating programs.

Firmware — there are three kinds of 'ware' in computers: software 'temporary' programs; hardware like the ROM which contains permanent information; and firmware in which the information is relatively permanent, as in an EPROM (see EPROM).

Flip-Flop — a circuit which maintains one electrical condition until changed to the opposite condition by an input signal.

FORTRAN — an acronym for FORmula TRANslation, this is a high level, problem orientated computer language for scientific and mathematical use.

G

Gate — an electrical circuit which, although it may accept one or more incoming signals, only sends out a single signal.

Graphics — pictorial information as opposed to letters and numbers.

H

Hard Copy — computer output which is in permanent form.

Hardware — the physical parts of the computer (also see software and firmware).

Hexadecimal (Hex) — a numbering system to the base sixteen. The digits zero to nine are used, as well as the letters A, B, C, D, E and F to represent numbers. A equals 10, B equals 11, C equals 12, and so on. Hex is often used by microprocessor users.

Hex Pad — a keyboard designed specifically for entering hexadecimal notation.

High Level Language — a programming language which allows the user to talk to the computer more or less in English. In general, the higher the level of the language (that is, the closer it is to English), the longer it takes for the computer to translate it into a language it can use. Lower level languages are far more difficult for human operators but are generally executed far more quickly.

I

Input — the information fed into the computer via a keyboard, a microphone, a cassette or a disk.

Input/Output (I/O Device) — a device which accepts information or instructions from the outside world, relays it to the computer, and then, after processing, sends the information out in a form suitable for storing, or in a form which could be understood by a human being.

Instruction — data which directs a single step in the processing of information by the computer (also known as a command).

Integrated Circuit — a complete electronic circuit imprinted on a semiconductor surface.

Interface — the boundary between the computer and a peripheral such as a printer.

Interpreter — a program which translates the high level language fed in by the human operator, into a language which the machine can understand.

Inverter — a logic gate that changes the signal being fed in, to the opposite one.

Interactive Routine — part of a program which is repeated over and over again until a specified condition is reached.

J

Jump Instruction — an instruction which tells the computer to go to another part of the program, when the destination of this move depends on the result of a calculation just performed.

K

K — this relates to the size of the memory. Memory is usually measured in 4K blocks. 1K contains 1,024 bytes.

Keyword — the trigger word in a line of programming, usually the first word after the line number. Keywords include STOP, PRINT and GOTO.

L

Language — computer languages are divided into three sections: high level languages, such as BASIC, which are reasonably close to English and fairly easy for humans to use; low level languages, such as Assembler, that use short phrases which have some connection with English (ADD for add and RET for return, for instance); and machine code which communicates more or less directly with the machine.

LCD — this stands for Liquid Crystal Diode. Some computers such as the TRS-80 Pocket Computer use an LCD display.

LED — this stands for Light Emitting Diode. The bright red numbers which are often used on watch or clock displays are made up of LEDs.

Logic — the mathematical form of a study of relationships between events.

Loop — a sequence of instructions within a program which is performed over and over again until a particular condition is satisfied.

M

Machine Language or Machine Code — an operation code which can be understood and acted upon directly by the computer.

Magnetic Disk — see Disk and Floppy Disk.

Mainframe — computers are generally divided into three groups, and the group a computer falls into depends more or less on its size. The computer you are thinking of buying is a microcomputer; medium sized computers are known as minicomputers; and the giant computers that you sometimes see in science fiction movies are mainframe computers. Until 15 years ago mainframe computers were, in practical terms, the only ones available.

Memory — there are two types of memory within a computer. The first is called ROM (read-only memory); this is the memory that comes already programmed on the computer, which tells the computer how to make decisions and how to carry out arithmetic operations. This memory is unaffected when you turn the computer off. The second type is RAM (random access memory). This memory holds the program you type in at the keyboard or send in via a cassette or disk. In most computers the computer 'forgets' what is in RAM when you turn the power off.

Microprocessor — the heart of any computer. It requires peripheral unit interfaces, such as a power supply and input and output devices, to act as a microcomputer.

MODEM — stands for Modulator Demodulator. This is a device which allows two computers to talk to each other over the telephone. The computers usually use a cradle in which a telephone receiver is placed.

Monitor — this has two meanings in computer terms. One meaning is a television-like display. A monitor has no facility for tuning television programs, and usually the picture produced on a monitor is superior to that produced by an ordinary television. The second meaning of a monitor relates to ROM. The monitor of a computer is described as the information it has built in when you buy it. This information allows it to make decisions and carry out arithmetic computations.

Motherboard — a framework to which extra circuits can be added. These extra circuits often give the computer facilities which are not built-in, such as that of producing sound or of controlling a light pen.

MPU — an abbreviation for Microprocessor Unit.

N

Nano-second — a nano-second is one thousand billionth of a second, the unit of speed in which a computer or a memory chip is often rated.

Non-Volatile Memory — memory which is not lost when the computer is turned off. Some of the smaller computers such as the TRS-80 Pocket Computer have non-volatile memory. The batteries hold the program you enter for several hundred hours.

Not — a Boolean logic operation that changes a binary digit into its opposite.

Null String — a string which contains no characters. It is shown in the program as two double quote marks, without anything between them.

Numeric — pertaining to numbers as opposed to letters (that is, alphabetic). Many keyboards are described as being alphanumeric which means both numbers and letters are provided.

O

Octal — a numbering system which uses eight as the base, and the digits 0, 1, 2, 3, 4, 5, 6 and 7. The Octal system is not used very much nowadays in microcomputer fields. The Hexadecimal system is more common (see Hexadecimal).

Operating System — the software or firmware generally provided with the machine that allows you to run other programs.

OR — an arithmetic operation that returns a 1, if one or more inputs are 1.

Oracle — a method of sending text messages with a broadcast television signal. A teletext set is required to decode the messages. Oracle is run by Independent Television Service in the UK, and a similar service — Ceefax — is provided by the BBC.

Output — information or data fed out by the computer to such devices as a TV-like screen, a printer or a cassette tape. The output usually consists of the information which the computer has produced as a result of running a program.

Overflow — a number too large or too small for the computer to handle.

P

Pad — see Keypad.

Page — often used to refer to the amount of information needed to fill one TV screen, so you can talk about seeing a page of a program, the amount of the listing that will appear on the screen at one time.

PASCAL — a high level language.

Peripheral — anything which is hooked onto a computer, for control by the computer, such as a disk unit, a printer or a voice synthesiser.

Port — a socket through which information can be fed out of or in to a computer.

Prestel — the British telecom name for a system of calling up pages of information from a central computer via the telephone and displaying them on a television screen. A similar commercial version in the United States is known as The Source.

Program — in computer terms program has two meanings. One is the list of instructions that you feed into a computer, and the second is used as a verb, as in 'to program a computer'.

PROM — stands for Programmable Read Only Memory. This is a device which can be programmed, and once it is then the program is permanent (also see EPROM and ROM).

R

Random Access Memory (RAM) — the memory within a computer which can be changed at will by the person using the computer. The contents of RAM are usually lost when a computer is turned off. RAM is the memory device that stores the program that you type in and also stores the results of calculations in progress.

Read-Only Memory (ROM) — in contrast to RAM, information in ROM cannot be changed by the user of the computer, and the information is not lost when the computer is turned off. The data in ROM is put there by the manufacturers and tells the computer how to make decisions and how to carry out arithmetic computations. The size of ROM and RAM is given in the unit K (see K).

Recursion — the continuous repetition of a part of the program.

Register — a specific place in the memory where one or more computer words are stored during operations.

Reserved Word — a word that you cannot use for a variable in a program because the computer will read it as something else. An example is the word TO. Because TO has a specific computer meaning, most computers will reject it as a name for a variable. The same goes for words like FOR, GOTO and STOP.

Routine — this word can be used as a synonym for program, or can refer to a specific section within a program (also see Subroutine).

S

Second Generation — this has two meanings. The first applies to computers using transistors, as opposed to first generation computers which used valves. Second generation can also mean the second copy of a particular program; subsequent generations are degraded by more and more noise.

Semiconductor — a material that is usually an electrical insulator but under specific conditions can become a conductor.

Serial — information which is stored or sent in a sequence, one bit at a time.

Signal — an electrical pulse which is a conveyor of data.

Silicon Valley — the popular name given to an area in California where many semiconductor manufacturers are located.

SNOBOL — a high level language.

Software — the program which is entered into the computer by a user which tells the computer what to do.

Software Compatible — this refers to two different computers which can accept programs written for the other.

Static Memory — a non-volatile memory device which retains information so long as the power is turned on, but does not require additional boosts of power to keep the memory in place.

Subroutine — part of a program which is often accessed many times during the execution of the main program. A subroutine ends with an instruction to go back to the line after the one which sent it to the subroutine.

T

Teletext — information transmitted in the top section of a broadcast television picture. It requires a special set to decode it to fill the screen with text information. The BBC service is known as Ceefax, the ITV service as Oracle. Teletext messages can also be transmitted by cable, for example the Prestel service in Britain or The Source in the United States.

Teletype — a device like a typewriter which can send information and also receive and print it.

Terminal — a unit independent of the central processing unit. It generally consists of a keyboard and a cathode ray display.

Time Sharing — a process by which a number of users may have access to a large computer which switches rapidly from one user to another in sequence, so each user is under the impression that he or she is the sole user of the computer at that time.

Truth Table — a mathematical table which lists all the possible results of a Boolean logic operation, showing the results you get from various combinations of inputs.

U

UHF — Ultra High Frequency (300-3000 megaHertz).

Ultra Violet Erasing — Ultra violet light must be used to erase EPROMs (see EPROM).

V

Variable — a letter or combination of letters and symbols which the computer can assign to a value or a word during the run of a program.

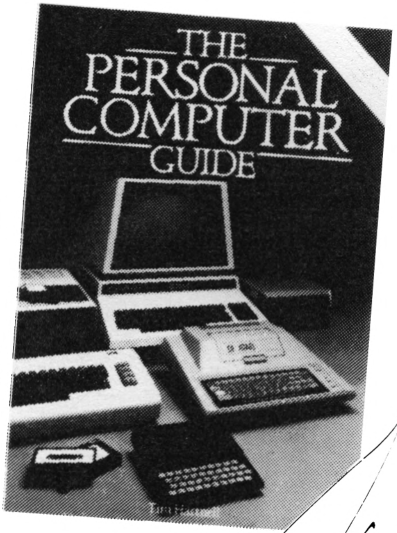
VDU — an abbreviation for Visual Display Unit.

Volatile — refers to memory which 'forgets' its contents when the power is turned off.

W

Word — a group of characters, or a series of binary digits, which represent a unit of information and occupy a single storage location. The computer processes a word as a single instruction.

Word-Processor — a highly intelligent typewriter which allows the typist to manipulate text, to move it around, to justify margins and to shift whole paragraphs if necessary on a screen before outputting the information onto a printer. Word-processors usually have memories, so that standard letters and the text of letters, written earlier, can be stored.


THE PERSONAL COMPUTER GUIDE

The Virgin Computer Games Series

GAMES FOR YOUR ZX81

More than 25 challenging programs, each one especially written for the series and guaranteed to provide hours of entertainment.

Moving graphics games, brain stretchers, word games and puzzles are all here. The games include TRAP (you're the fugitive, the computer is your pursuer); TANK ASSAULT (a fast-moving game in which you manoeuver through a minefield); STAR TRADER (become a galactic entrepreneur); POKER (how to lose a fortune painlessly); CAVERNS (a test of mental arithmetic); GOLF (win the Open without moving from your sitting-room); and POLOMONY (build an empire in an evening).

GAMES FOR YOUR ZX81 will improve your programming skills as you follow the instructions to put each of the programs into your machine, and comes complete with a brief dictionary of computer terms and some hints on how to improve and extend the programs in the book.

Programs of originality and quality for all the family.

Virgin

0 907080 82 0

£2.95

THE VIRGIN COMPUTER
GAMES SERIES

GAMES FOR YOUR ZX81

Virgin